CLaSSiC Project
Manuscript CLaSSiC–91–29

August 1991

Numerical Computation of Unsteady Incompressible Flow in Complex Geometry Using a Composite Multigrid Technique

M. Hinatsu
Joel Ferziger

Center for Large Scale Scientific Computation
Building 460, Room 313
Stanford University
Stanford, California 94305

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION: UNCLASSIFIED	1b. RESTRICTIVE MARKINGS: NONE
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT: Unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S): CLaSSiC Manuscript 91-29	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION: Stanford University; 6b. OFFICE SYMBOL (If applicable): 2E254	7a. NAME OF MONITORING ORGANIZATION: Department of the Navy, Office of Naval Research
6c. ADDRESS (City, State, and ZIP Code): c/o Sponsored Projects Office, Encina Hall, 660 Arguello Way, Stanford, CA 94305	7b. ADDRESS (City, State, and ZIP Code): 800 North Quincy Street, Arlington, VA 22217-5000
8a. NAME OF FUNDING/SPONSORING ORGANIZATION: Office of Naval Research; 8b. OFFICE SYMBOL (If applicable): N00014	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER: N00014-90-J-1344
8c. ADDRESS (City, State, and ZIP Code): 800 North Quincy Street, Arlington, VA 22217-5000	10. SOURCE OF FUNDING NUMBERS: PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO / WORK UNIT ACCESSION NO

11. TITLE (Include Security Classification): Numerical Computation of Unsteady Incompressible Flow in Complex Geometry Using a Composite Multigrid Technique

12. PERSONAL AUTHOR(S): M. Hinatsu and Joel Ferziger

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Interim	FROM 90Sep01 TO 91Jan3	1991 August	48

16. SUPPLEMENTARY NOTATION

17. COSATI CODES			18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
FIELD	GROUP	SUB-GROUP	Unsteady Navier-Stokes, composite multigrid method, fractional step method, staggered grid

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

This paper presents a composite multigrid method and its application to a geometrically complex flow. The treatment of the interior boundary boundary conditions within a composite multigrid strategy is described in detail for a 1-D model equation. For the Navier-Stokes equations, a staggered grid technique is adopted for spatial discretization and a fractional step method is used for the time advance. Lid driven cavity flows are used to demonstrate the effectiveness of the method.

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT: ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION: UNCLASSIFIED
22a. NAME OF RESPONSIBLE INDIVIDUAL: Joel Ferziger	22b. TELEPHONE (Include Area Code): (415) 723-3615; 22c. OFFICE SYMBOL

DD Form 1473, JUN 86 — Previous editions are obsolete. — SECURITY CLASSIFICATION OF THIS PAGE

S/N 0102-LF-014-6603

UNCLASSIFIED

NUMERICAL COMPUTATION OF UNSTEADY INCOMPRESSIBLE FLOW IN COMPLEX GEOMETRY USING A COMPOSITE MULTIGRID TECHNIQUE

M. HINATSU* and J.H.FERZIGER**

January 3, 1991

* Ship Propulsion Division, Ship Research Institute
Ministry of Transport of Japan
6-38-1, Shinkawa, Mitaka, Tokyo, 181, Japan
** Thermosciences Division, Department of Mechanical Engineering
Stanford University, Stanford, California, 94305, USA

SUMMARY

This paper presents a composite multigrid method and its application to a geometrically complex flow. The treatment of the interior boundary boundary conditions within a composite multigrid strategy is described in detail for a 1-D model equation. For the Navier-Stokes equations, a staggered grid technique is adopted for spatial discretization and a fractional step method is used for the time advance. Lid driven cavity flows are used to demonstrate the effectiveness of the method.

KEY WORDS

Unsteady Navier-Stokes, composite multigrid method, fractional step method, staggered grid.

1. INTRODUCTION

Many methods for numerical simulation of fluid flows have been proposed and CFD (Computational Fluid Dynamics) has become a practical design tool. However, problems remain with regard to computational efficiency, accuracy,

and turbulence modeling. The most difficult problem may be mesh generation for geometrically complicated domains.

Many schemes have been devised to cope with these issues. The multigrid method [1] is one of the most efficient schemes for elliptic problems and has been applied to the Navier-Stokes equations. To deal with geometric complexity, domain decomposition has been proposed: its origins go back to Schwarz [2]. The idea is that a complicated domain be decomposed into subdomains, whose geometry is simple enough to be easily gridded.

In the present work, we apply a combination of these two methods, that is, a composite multigrid strategy, to flows in geometrically complicated domains. In particular, we treat 2-D unsteady laminar flows. The Navier-Stokes equations are discretized using second order central differencing on a staggered grid in space and a fractional step time advance method. The velocity components are advanced explicitly and the pressure is obtained by solving a Poisson equation using a composite multigrid method. The momentum equations are integrated independently on each subgrid. Interpolation on the composite grid is accomplished with a Coons patch method [3].

Prior to solving the Navier-Stokes equations, we investigate the effectiveness of the composite multigrid technique for a 1-D model equation. We also discuss the interior boundary conditions.

After demonstrating the properties of the composite multigrid method, we appply it to the Navier-Stokes equations. First, we check the accuracy of the method for lid-driven cavity flow at $Re = 3200$ and evaluate the error. Then, we simulate geometrically complex cavity flows.

2. THE GOVERNING EQUATIONS

The 2-D unsteady incompressible Navier-Stokes (N-S) equations can be written in the following form in a curvilinear coordinate system.

$$\frac{\partial \hat{q}}{\partial t} + \frac{\partial E}{\partial \xi} + \frac{\partial F}{\partial \eta} + \frac{\partial H_\xi}{\partial \xi} + \frac{\partial H_\eta}{\partial \eta} + \frac{\partial P}{\partial \xi} + \frac{\partial Q}{\partial \eta} = 0 \tag{1}$$

where

$$\hat{q} = q/J, \quad q = [u, v]^T, \quad J = \xi_x \eta_y - \xi_y \eta_x \tag{2}$$

and u and v are the Cartesian velocity components along the x and y axes and ξ and η are arbitrary curvilinear coordinates. J is the Jacobian. Subscripts x and y stand for derivatives with respect to x and y.
The second and third terms of LHS (1) are the convection terms so that E and F are

$$E = U\hat{q}, \quad F = V\hat{q} \tag{3a}$$

where

$$U = u\xi_x + v\xi_y, \quad V = u\eta_x + v\eta_y. \tag{3b}$$

are the contravariant velocity components; however, we shall use the Cartesian velocity components as the primary variables.
The fourth and fifth terms are the pressure gradient terms; H_ξ and H_η can be written as:

$$H_\xi = \begin{pmatrix} H_{\xi u} \\ H_{\xi v} \end{pmatrix} = \begin{pmatrix} \xi_x p/J \\ \xi_y p/J \end{pmatrix}, \quad H_\eta = \begin{pmatrix} H_{\eta u} \\ H_{\eta v} \end{pmatrix} = \begin{pmatrix} \eta_x p/J \\ \eta_y p/J \end{pmatrix} \tag{4}$$

where p is the pressure.
The last two terms are the viscous terms; P and Q have the form,

$$P = -\frac{1}{J \cdot Re} \begin{pmatrix} (\xi_x^2 + \xi_y^2)u_\xi + (\xi_x\eta_x + \xi_y\eta_y)u_\eta \\ (\xi_x^2 + \xi_y^2)v_\xi + (\xi_x\eta_x + \xi_y\eta_y)v_\eta \end{pmatrix}$$

$$Q = -\frac{1}{J \cdot Re} \begin{pmatrix} (\eta_x^2 + \eta_y^2)u_\eta + (\xi_x\eta_x + \xi_y\eta_y)u_\xi \\ (\eta_x^2 + \eta_y^2)v_\eta + (\xi_x\eta_x + \xi_y\eta_y)v_\xi \end{pmatrix} \tag{5}$$

where Re is the Reynolds number.
The above equations have to be solved simultanously with the continuity equation:

$$\frac{1}{J}\left(\frac{\partial}{\partial\xi}\left(\xi_x\frac{u}{J}+\xi_y\frac{v}{J}\right)+\frac{\partial}{\partial\eta}\left(\eta_x\frac{u}{J}+\eta_y\frac{v}{J}\right)\right) = \frac{1}{J}\left(\frac{\partial}{\partial\xi}\left(\frac{U}{J}\right)+\frac{\partial}{\partial\eta}\left(\frac{V}{J}\right)\right) = 0. \tag{6}$$

3. NUMERICAL SCHEME

3.1 Differencing in space and time

The momentum equations for the velocity components u and v can be written

$$\frac{\partial}{\partial t}\left(\frac{u}{J}\right) + L^u(u, v, u_\xi, u_\eta, v_\xi, v_\eta, p) = 0 \tag{7}$$

$$\frac{\partial}{\partial t}\left(\frac{v}{J}\right) + L^v(u, v, u_\xi, u_\eta, v_\xi, v_\eta, p) = 0 \tag{8}$$

where L^u and L^v are the sum of the convection, pressure gradient, and viscous diffusion operators.
Multiplying Eq (7) by ξ_x or η_x and multiplying Eq (8) by ξ_y or η_y and summing, we get the equations for U and V.

$$\frac{\partial}{\partial t}\frac{1}{J}\begin{pmatrix} U \\ V \end{pmatrix} + \begin{pmatrix} \xi_x \\ \eta_x \end{pmatrix} L^u(u, v, \cdots, p) + \begin{pmatrix} \xi_y \\ \eta_y \end{pmatrix} L^v(u, v, \cdots, p) = 0 \tag{9}$$

To discretize this equation in space, we use second order central differencing on a staggered grid in which the pressure node is located at the cell center and the contravariant velocity components U and V are located on the cell

boundaries as shown in Fig. 1. Control volume I is used for the x-momentum equation and control volume II for the y-momentum equation. In order to evaluate u and v at the velocity nodes, we must evaluate U and V at those nodes, but, on a staggered grid, U and V are known at different points. Maliska and Raithby [4] used the average of the values at four surrounding points to obtain the velocity component on a node of the other component. For example, to evaluate V at the point A of Fig. 1, they took an average of V at B, C, D and E. We adopt this method.

After evaluating U and V, we get u and v from the following relation.

$$\begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} \xi_x & \xi_y \\ \eta_x & \eta_y \end{pmatrix}^{-1} \begin{pmatrix} U \\ V \end{pmatrix} \tag{10}$$

For discretization of the convection terms, we use the QUICK scheme [5].

Time advance is carried out using the fractional step method. Let u^n, v^n and p^n be the velocity and pressure at time step n and assume that u^n and v^n satisfy the continuity equation.

First, we evaluate $U^{\overline{n+1}}$ and $V^{\overline{n+1}}$ by explicit integration of Eq (9).

$$\frac{1}{J}\begin{pmatrix} U \\ V \end{pmatrix}^{\overline{n+1}} = \frac{1}{J}\begin{pmatrix} U \\ V \end{pmatrix}^{n} - \Delta t\left(L^u(u^n, v^n, \ldots, p^n)\begin{pmatrix} \xi_x \\ \eta_x \end{pmatrix} + L^v(u^n, v^n, \ldots, p^n)\begin{pmatrix} \xi_y \\ \eta_y \end{pmatrix}\right) \tag{11}$$

This is the predictor of the MacCormack scheme [6] and many others.

The corrector for the MacCormack scheme is

$$\frac{1}{J}\begin{pmatrix} \tilde{U} \\ \tilde{V} \end{pmatrix} = \frac{1}{2}\left[\frac{1}{J}\begin{pmatrix} U \\ V \end{pmatrix}^{n} + \frac{1}{J}\begin{pmatrix} U \\ V \end{pmatrix}^{\overline{n+1}} - \Delta t\left(L^u(u^{\overline{n+1}}, v^{\overline{n+1}}, \ldots, p^n)\begin{pmatrix} \xi_x \\ \eta_x \end{pmatrix} + L^v(u^{\overline{n+1}}, v^{\overline{n+1}}, \ldots, p^n)\begin{pmatrix} \xi_y \\ \eta_y \end{pmatrix}\right)\right] \tag{12}$$

where $u^{\overline{n+1}}, v^{\overline{n+1}}$ are computed from $U^{\overline{n+1}}, V^{\overline{n+1}}$ using Eq (10).

Using Eq (11), Eq (12) can be rewritten as

$$\frac{1}{J}\begin{pmatrix} \tilde{U} \\ \tilde{V} \end{pmatrix} - \frac{1}{J}\begin{pmatrix} U \\ V \end{pmatrix}^{n} = -\frac{1}{2}\Delta t\left[L^u(u^n, v^n, \ldots, p^n)\begin{pmatrix} \xi_x \\ \eta_x \end{pmatrix} + L^v(u^n, v^n, \ldots, p^n)\begin{pmatrix} \xi_y \\ \eta_y \end{pmatrix} + L^u(u^{\overline{n+1}}, v^{\overline{n+1}}, \ldots, p^n)\begin{pmatrix} \xi_x \\ \eta_x \end{pmatrix} + L^v(u^{\overline{n+1}}, v^{\overline{n+1}}, \ldots, p^n)\begin{pmatrix} \xi_y \\ \eta_y \end{pmatrix}\right] \tag{13}$$

The new velocity field $\tilde{u}, \tilde{v}$ does not satisfy the continuity equation (10) so we introduce a pressure correction and compute the new velocity field U^{n+1}, V^{n+1}

from

$$\frac{1}{J}\begin{pmatrix} U^{n+1} \\ V^{n+1} \end{pmatrix} - \frac{1}{J}\begin{pmatrix} \tilde{U} \\ \tilde{V} \end{pmatrix} = -\left(\frac{\partial}{\partial \xi}(\Delta H_{\xi u}) + \frac{\partial}{\partial \eta}(\Delta H_{\eta u})\right)\begin{pmatrix} \xi_x \\ \eta_x \end{pmatrix} - \left(\frac{\partial}{\partial \xi}(\Delta H_{\xi v}) + \frac{\partial}{\partial \eta}(\Delta H_{\eta v})\right)\begin{pmatrix} \xi_y \\ \eta_y \end{pmatrix} \tag{14}$$

where $\Delta H_{\xi u} = H^{n+1}_{\xi u} - H^{n}_{\xi u}$, *etc.* are determined by forcing U^{n+1} and V^{n+1} to satisfy the continuity equation. In this way, we obtain a pressure equation (strictly speaking, a pressure increment equation).

$$\frac{\partial}{\partial \xi}\left[\xi_x\left(\frac{\partial}{\partial \xi}(\Delta H_{\xi u}) + \frac{\partial}{\partial \eta}(\Delta H_{\eta u})\right) + \xi_y\left(\frac{\partial}{\partial \xi}(\Delta H_{\xi v}) + \frac{\partial}{\partial \eta}(\Delta H_{\eta v})\right)\right] + \frac{\partial}{\partial \eta}\left[\eta_x\left(\frac{\partial}{\partial \xi}(\Delta H_{\xi u}) + \frac{\partial}{\partial \eta}(\Delta H_{\eta u})\right) + \eta_y\left(\frac{\partial}{\partial \xi}(\Delta H_{\xi v}) + \frac{\partial}{\partial \eta}(\Delta H_{\eta v})\right)\right] = \frac{\partial}{\partial \xi}\left(\frac{\tilde{U}}{J}\right) + \frac{\partial}{\partial \eta}\left(\frac{\tilde{V}}{J}\right) \tag{15}$$

This is a Poisson equation for $\Delta p = p^{n+1} - p^n$ and can be solved by the multigrid method. Because we use a staggered grid, no boundary condition is needed for the pressure equation provided the coordinate system is orthogonal at the boundary; this is the case in the present work. However, the discrete system of equations is singular. For a simplicity of coding, we use a fictitious pressure node outside the computational domain to eliminate the singularity. Furthermore, to ensure uniqueness, we set the solution at an arbitrary point (for instance, at $(\xi, \eta) = (1/2, 1/2)$) to zero. Other methods of de-singularizing the system are available.

3.2 Multigrid method for the pressure

Multigrid [1] is a well known method for solving elliptic equations. We apply it only to the pressure equation. The essential idea of the multigrid method is reduction of the error using several grids of different sizes; high frequency components of the error are removed by smoothing on a fine grid and low frequency components are damped on coarser grids.

The system of linear equations (15) can be written

$$L\phi = f \tag{16}$$

where ϕ is the solution and f is the source (forcing) term.

After we relax Eq (16) using suitable smoother on the finest grid, we get an approximate solution $\tilde{\phi}^k$ and

$$L^k \tilde{\phi}^k = f - R^k \tag{17}$$

where R is the residual and superscript k stands for the grid level; $k = k_{max}$ corresponds to the finest grid and $k-1$, a grid twice as coarse. The smoother should reduce the high frequency components of the error rapidly. Subtracting Eq (17) from Eq (16), we get an equation for the error $\tilde{e}^k$:

$$L^k \tilde{e}^k = R^k \quad , \quad \tilde{e}^k = \phi - \tilde{\phi}^k \tag{18}$$

Next, we remove lower frequency components of $\tilde{e}$. To do that, Eq (18) is solved on grid $k-1$.

$$L^{k-1}\tilde{e}^{k-1} = R^{k-1} \quad , \quad R^{k-1} - I_k^{k-1} R^k \tag{19}$$

where I_k^{k-1} is a restriction operator.
After smoothing $\tilde{e}$ on the coarse grid, $\tilde{e}$ is interpolated onto the fine grid by interpolation or prolongation:

$$\tilde{e}^k = I_{k-1}^k \tilde{e}^{k-1} \tag{20}$$

where I_{k-1}^k is an interpolation operator.
The solution is then updated:

$$\phi^k = \phi_{old}^k + \tilde{e}^k \tag{21}.$$

In the present work, since we use a staggered grid system, I_k^{k-1} and I_{k-1}^k are defined as follows.

$$I_k^{k-1}\phi_{ic,jc}^k = (\phi_{if,jf}^k + \phi_{if+1,jf}^k + \phi_{if,jf+1}^k + \phi_{if+1,jf+1}^k)/4 \tag{22}$$

where $if = 2ic - 1$ and $jf = 2jc - 1$

$$I_{k-1}^k \phi_{if,jf}^k = (9\phi_{ic,jc}^k + 3\phi_{ic+1,jc}^k + 3\phi_{ic,jc+1}^k + \phi_{ic+1,jc+1}^k)/16 \tag{23}$$

where $ic = 1 + if/2$ and $jc = 1 + jf/2$ (see Fig. 2).

The method is readily extended to larger numbers of grids.

4. DOMAIN DECOMPOSITION TECHNIQUE

In order to solve the Navier-Stokes equations, we have to divide the domain using a suitable grid system. For flow fields that do not have simple geometry, covering the entire domain with a single grid is difficult.

Domain decomposition is a method of coping with this problem. In this technique, a geometrically complicated domain is divided into several simpler ones. Schwarz [2] proposed an alternating solution procedure for elliptic problems. Since this method uses Dirichlet conditions on the interior boundaries, it requires that the subdomains overlap.

Van der Wijngaart [7] revised the Schwarz method using asymmetric interior boundary conditions. If Neumann conditions are used for the interior

boundary condition of Grid I, Dirichlet conditions are applied on Grid II. He showed that this treatment allows the requirement of subdomain overlap to be removed. More recently, Lions [8] developed a theory of the Schwarz method on non-overlapping subdomains.

A few applications of this method to fluid flow simulation exist. Meakin and Street [9] simulated a 3-D environmental flow using a composite grid method. Van der Wijngaart [7] developed the SWAPR (Schwarz Alternating Procedure Revised) method which uses asymmetric boundary conditions on the interior boundary.

In the present work, we combine domain decomposition with the multigrid method. Henshaw and Chesshire [10] solved the Poisson equation using a composite multigrid technique. Perng [11] and Perng and Street [12] simulated flows in complicated domains using a combination of explicit time advance on individual grids and a multigrid pressure solver on the complete composite grid. They showed the effectiveness of the composite multigrid method for geometrically complicated flows including 3-D problems. However, they restricted the domain decomposition by requiring that grid nodes in overlapping domains belong to both grids. This restriction reduces freedom in grid construction and is rmoved in the present work. On the interior boundaries, Perng and Street adopted Neumann boundary conditions but also used Dirichlet conditions.

Before describing the Navier-Stokes solver, we consider the composite multigrid method for a simple 1-D model problem.

4.1 Multigrid composite grid technique (1-D model problem)

Consider the following 1-D model problem.

$$y'' = 2, \quad (0 \leq x \leq 1), \quad \text{with} \quad y(0) = 0, \quad y'(1) = 1 \tag{24}$$

The exact solution is

$$y = x(x-1) \tag{25}$$

We choose two overlapping grids, one from $x = 0$ to $x = 0.6$ and the other covering $x = 0.4$ to $x = 1$; we call the former, Grid I and the latter, Grid II. Each grid is divided into 16 equal elements, so the finest grid size is 0.0375. Three levels are used on each grid; the finest has 16 intervals and the coarsest, 4 intervals. In this 1-D problem, we adopt a regular grid, so the coarse grid nodes coincide with the fine grid nodes.

To use Schwarz iteration, we have to estimate the interior boundary value by interpolating from the other grid. Two solution methods are considered. In the first, interior boundary value communication is limited to the finest grid and the coarse grid smoothings are carried out independently. In the second method, data communication at the interior boundary is allowed at every grid

level. Hereafter, we call the former the incomplete composite multigrid (or simply ICMG) method and the latter the complete composite multigrid (or CCMG) method. Henshaw et al. [10] used the CCMG method while Perng and Street [12] used the ICMG method.

The algorithms for two method are as follows.

a) ICMG method

i) Iterate the equation on Grid I using a local V-cycle with a guessed interior boundary value.
ii) The interior boundary value for the finest level of Grid II is obtained by interpolation on Grid I.
iii) Iterate the equation on Grid II using a local V-cycle.
iv) Find the boundary condition at the finest level of Grid I by interpolation on Grid II.
v) Repeat (i) - (iv) until the solution converges.

b) CCMG method

i) Iterate the equation on the finest level of Grid I using a guessed interior boundary value.
ii) The interior boundary condition for the finest level of Grid II found by interpolation on Grid I.
iii) Iterate the equation on the finest level on Grid II.
iv) Interpolate to find the Grid I boundary condition.
v) Repeat (i)-(iv) (Schwarz iteration on the finest grid level). There is no need to iterate to convergence. In the test computation, we iterated 2 times.
vi) The residual on each composite grid is restricted to the coarser grid and the correction is iterated using procedures (i) to (v).
vi) After solving at the coarsest grid level, work back to the finer grids.
vii) Iterate procedures i) to vi) (V-cycle) until the solution converges.

4.2 Interior boundary condition for multigrid composite grid technique

At the interior boundaries, we tried two types of conditions, Neumann and Dirichlet.

First, consider the ICMG method. In this case, data communication is limited to the finest grid and the interior boundary condition is found as follows.

For the Dirichlet boundary condition case,

$$y^I_{right} = ITP(y^{II}) \quad , \quad y^{II}_{left} = ITP(y^I) \tag{26}$$

where ITP is an interpolation operator and y^I means the solution on Grid I and y^I_{right} is the right boundary value on Grid I.

We can also introduce an overrelaxation parameter to accelerate convergence as was done by Tang [13].

$$y^I_{right} = \omega ITP(y^{II}) + (1-\omega)(y^I_{right})_{old} \tag{27}$$

In a similar manner, we find y^{II}_{left}

The case of Neumann boundary conditions is similar.

$$dy^I_{right}/dx = ITP(dy^{II}/dx)$$

or

$$dy^I_{right}/dx = \omega ITP(dy^{II}/dx) + (1-\omega)(dy^I_{right}/dx)_{old} \tag{28}$$

In the ICMG method, relaxation on each set of coarse grids is carried out independently, thus the interior boundary condition for the error is

$$\begin{aligned} \tilde{e}^I_{right} &= 0 \quad \text{Dirichlet} \\ d\tilde{e}^I_{right}/dx &= 0 \quad \text{Neumann} \end{aligned} \tag{29}$$

Next, consider the CCMG method. The boundary condition on the finest grid level is found as in the ICMG case. On the coarser grids, the treatment of the boundary condition is somewhat more complicated.

After iterating the correction equation on the coarse grid, the result satisfies the following relation.

$$\begin{aligned} (y^I_{new})_{right} &= ITP(y^{II}_{new}) \quad \text{Dirichlet} \\ (dy^I_{new}/dx)_{right} &= ITP(dy^{II}_{new}/dx) \quad \text{Neumann} \end{aligned}$$

In other words, instead of Eq (29), the boundary condition on the error on the interior boundary should be

$$\begin{aligned} (y^I_{old} + \tilde{e}^I)_{right} &= ITP(y^{II}_{old}) + ITP(\tilde{e}^{II}) \quad \text{Dirichlet} \\ (dy^I_{old}/dx + d\tilde{e}^I/dx)_{right} &= ITP(dy^{II}_{old}/dx) + ITP(d\tilde{e}^{II}/dx) \quad \text{Neumann} \end{aligned} \tag{30}$$

Again, we can accelerate the boundary conditions:

$$\tilde{e}^I = \omega \tilde{e}^* + (1-\omega)\tilde{e}^I_{old}$$

where $\tilde{e}^* = ITP(y^{II}_{old}) + ITP(\tilde{e}^{II}) - y^I_{old}$. The Neumann boundary condition can be treated in a similar manner.

Although Eq (30) is the exact boundary condition for the error, we may approximate it in a way that makes data communication at each grid level independent, namely

$$\tilde{e}^I_{right} = ITP(\tilde{e}^{II}) \quad or \quad d\tilde{e}^I_{right}/dx = ITP(d\tilde{e}^{II}/dx). \tag{31}$$

In the present study, we tested both Eq (30) and Eq (31).

4.3 Remarks on the 1-D model problem

Next, we describe some 1-D test computations. The relaxation method was Gauss-Seidel. Two sweeps were performed at each grid level. On the coarsest grid, iteration was carried to convergence. For the ICMG method, one V-cycle was made on each subgrid and the total number of Schwarz iterations was limited to 20. For the CCMG method, two Schwarz iterations were performed at each grid level and ten V-cycles were carried out. Thus, the work is almost same for the two methods. Because the CCMG method requires data communication on each grid level, its cost is slightly larger.

Now we present results for the 1-D test problem. The test computations are (a) the ICMG method with Dirichlet interior boundary conditions (i.b.c.), (b) the ICMG method with Neumann i.b.c., (c) the CCMG method with Dirichlet i.b.c. (Eq (30)), (d) the CCMG method with Dirichlet i.b.c. (Eq (31)), (e) the CCMG method with Neumann i.b.c. (Eq (30)) and (f) the CCMG method with Neumann i.b.c. (Eq (31)). We show the normalized errors after 20 Schwarz (1 V-cycle) iterations for the ICMG method and after 2 Schwarz (10 V-cycle) iterations for the CCMG method to judge the rapidity of convergence. The error is defined by

$$error = \sqrt{\frac{\sum (y_{numerical} - y_{exact})^2}{\sum y_{exact}^2}} \tag{32}$$

The errors are given as functions of the acceleration parameter ω. Fig. 3 shows the errors with the Dirichlet i.b.c. for the ICMG and CCMG methods and Fig. 4 gives the errors with Neumann i.b.c. for the same methods. Error I is the error on the finest level of subgrid I. In both figures, results obtained with both Eqs (30) and (31) are shown for the CCMG method. The ICMG method converges faster with Neumann i.b.c. than with Dirichlet i.b.c.. In both cases, the convergence is accelerated by overrelaxation of the interior boundary values. The optimal relaxation parameters are $\omega \simeq 1.6$ in the Dirichlet i.b.c. case and $\omega \simeq 1.3$ in the Neumann i.b.c. case.

The ICMG method converges faster than the CCMG method, especially with Dirichlet i.b.c.s. For both types of i.b.c.s, Eq (30) gives faster convergence than does Eq (31); the difference in CPU time per time step is very slight so the method based on Eq (30) is faster overall. Interior boundary value acceleration is less effective in the CCMG method. With Neumann i.b.c., the best convergence in the CCMG method is obtained with $\omega = 1$.

However, these trends change with the external boundary conditions at $y = 0$ and 1. For instance, when Dirichlet boundary conditions are given, i.e., $y(0) = y(1) = 0$, with Dirichlet i.b.c.s, the error as a function of ω is shown in Fig. 5. The CCMG method converges much faster than in the previous

case. However, for optimal ω (in this case $\omega = 1.5$), the ICMG method remains faster.

The convergence rate also depends on the size of the overlap region. For these tests, the external boundary conditions were $y(0) = 0, \quad y'(1) = 1$ and Neumann i.b.c. were used. Test computations were made with 20%, 30%, 40% and 50% overlap. Fig. 6 shows the error for the ICMG method after 20 Schwarz iterations as a function of the overlap. The number of grid points and the number of levels are the same as before, 16 equal elements and 3 multigrid levels. The optimal acceleration parameter is only weakly affected by the amount of overlap, but the convergence is much faster when the overlap is large.

Finally, we compared the convergence speed between a single (non-composite) grid and a composite grid. We solved Eq (24) using 32-point single grid and a 50% overlap composite grid. The finest level of the composite grid has 24 intervals of the same size as the finest non-composite grid. Three levels were used in both cases. The ICMG method was used on the composite grid with $\omega = 1.2$. Fig. 7 shows the maximum residual on each subgrid. Since we must guess the initial interior boundary conditions for the ICMG method, the residual at the first iteration is much bigger than in non-composite grid method. This requires the ICMG method to take more iterations. However, the convergence rate is almost same so the cost increase is almost entirely due to the overlap.

4.4 Composite multigrid method for a 2-D model problem

The method can be applied to 2-D boundary value problems. Since we intend to apply the method to the pressure equation, the Poisson equation was chosen as a test problem. In the test computation, the domain consisted of two squares shifted diagonally by 40% of the diagonal length. The physical domain and the mesh are shown in Fig. 8.

Neumann boundary conditions were applied at the entire boundary of the computational domain. We introduced four singularities as the forcing; their strengths add to zero for consistency with the boundary condition $\partial\phi/\partial n = 0$.

$$\int_S \nabla^2\phi \, dS = \int_S f \, dS = \int_L \frac{\partial\phi}{\partial n} \, dl = 0 \tag{33}$$

For the interior boundary condition, we can use Dirichlet or Neumann conditions. Because the fine grid nodes do not coincide with nodes of the coarse grid, if we use Dirichlet boundary conditions on a staggered grid, it is difficult to give boundary conditions for the error. Also, because Neumann boundary conditions are applied on the physical boundary, the programming is simpler if Neumann i.b.c.s are used. Finally, as shown in the 1-D test problem, Neumann i.b.c.s give faster convergence so this is the choice we shall make.

Now let us describe the implementation of the interior Neumann boundary conditions. As mentioned in Section 2, we used fictitious points outside the

computational domain. In Fig. 9, suppose points A and B are members of Grid I contained in Grid II. In the ICMG method, values at A and B are obtained by interpolating Grid II data. The boundary condition for the error requires its gradient to be zero.

In the CCMG method, we need coarse grid boundary information as well. To find the boundary condition at point E in Fig. 9, the error at points F and G is interpolated. Further, the solution on the fine grid at points A, B, C and D is interpolated to give the derivative at points H and I.

For interpolation, we used the Coons patch technique [3] with second order accuracy. Although analysis indicates that a third order method should be used, we find that this method is sufficiently accurate; it does, however, lead to some minor discrepancies in the results that will be discussed below. Fig. 10 shows the stencil for second order Coons patch interpolation. P in Fig. 10 is the point at which interpolated data are required; α and β are the local coordinates of P. The interpolated data can be expressed as

$$\phi(P) = (1-\alpha)\mathbf{D} + \alpha\mathbf{B} + (1-\beta)\mathbf{A} + \beta\mathbf{C}$$
$$-(1-\alpha)(1-\beta)\phi(a) - (1-\beta)\alpha\phi(b) - \alpha\beta\phi(c) - (1-\alpha)\beta\phi(d) \qquad (34)$$

where $\phi(a)$ is the value at the corner a. The local coordinates (α, β) of the four corners are $a = (0,0), b = (1,0), c = (1,1)$ and $d = (0,1)$. $\mathbf{A}, \mathbf{B}, \mathbf{C}$ and $\mathbf{D}$ are second order polynomials along the edges ab, bc, cd and da, respectively (see Fig. 10).

We now show the results of a test computation. In Fig. 11, the converged solution is shown. The solutions completely coincide in the overlap domain. In Fig. 12, the maximum residual in each subgrid is shown as a function of iteration number. The fastest convergence is obtained with the ICMG method with $\omega = 1.1$. As in the 1-D problem, the effect of ω on convergence is much weaker for the CCMG method than for the ICMG method. Of the interior boundary conditions applied to the CCMG method, Eq. (30) yields the fastest convergence. The ICMG method converges faster than the CCMG method. Thus we used the ICMG method in the Navier- Stokes solver.

To investigate the speed of the composite grid method relative to a single grid method, we solved the four-singularity problem in a square domain. 10 V-cycle iterations were performed for the single grid method and 10 Schwarz 1 V-cycle iterations were carried out for the ICMG method. Fig. 13(a) shows the result obtained using a single (33 x 33) grid and Fig. 13(b) shows the result using the ICMG method. The composite grids are each 33 x 25 and, in the overlap region, two subgrids coincide. The two solutions agree. The single grid method required 29.1sec on a SUN-3 with a floating point accelerator and the ICMG method required 61.5sec. Computations were performed using double precision. The maximum residual after 10 V-cycle computations on the single grid is 4.38×10^{-6}. For the ICMG method, the maximum residuals on the two grids after 10 Schwarz iterations are 9.43×10^{-6} and 4.20×10^{-6} and maximum

difference of the gradient on the interior boundary is 1.17×10^{-8}.

5. COMPOSITE MULTIGRID METHOD FOR THE N-S EQUATIONS

The composite multigrid method is used to solve the pressure equation. The momentum equations are updated explicitly on the finest grid. Because we use the QUICK scheme, we need two fictitious points outside the boundaries; the velocity components are obtained by interpolation using Coons patch at these points (see Fig. 14). There is a difference between the stencils for pressure and velocity interpolation; it is shown in Fig. 15. For pressure interpolation, the stencil contains nine cells, while for velocity interpolation, only one cell used. Velocity data at the cell corners are found by extrapolating the adjacent velocity data.

6. CHECK OF NUMERICAL SCHEME OF N-S SOLVER

In order to check the computer program, we computed lid driven cavity flow at Re=3200. First, we used a single grid. The computations were carried out on 33×33, 41×41, 49×49 and 65×65 grids with $\Delta t = 0.02, 0.015, 0.015$ and 0.01, respectively. The convergence criterion required the maximum increment of velocity in one time step to be less than 1×10^{-6}. A CYDRA-5 mini-supercomputer was used; for the 33×33 case, 9500 time steps and 160 minutes of CPU time were required to reach steady state. Except in the early stages of the calculation, for which the pressure solver was limited to 5 V-cycles, we required that the maximum change in the pressure between iterations be less that 1×10^{-8}.

The convergence error can be estimated as [14]

$$\epsilon^n \simeq \frac{\phi^{n+1} - \phi^n}{\lambda_1 - 1}. \tag{35}$$

where λ_1 is the largest eigenvalue (spectral radius) of the iteration matrix. In the Poisson solver with the Gauss-Seidel method, it can be approximated by $\lambda_1 \simeq \pi^2/N^2$, N being the number of grid points. For the momentum equations, we can not evaluate λ_1 in advance but, if the maximum eigenvalue is real, we can use the estimate [15]

$$\lambda_1 \simeq \frac{\Delta u^{n+1}}{\Delta u^n} \tag{36}$$

The results show that $\lambda_1 - 1 \simeq O(10^{-3})$. Thus, we estimate the convergence error as 10^{-3}.

Since the present scheme has second order accuracy in space, the solution can be expanded in the Taylor series about the exact solution.

$$\phi = \phi_{exact} + a(\Delta x)^2 + b(\Delta x)^3 + \cdots \tag{37}$$

On a sufficiently fine mesh the error should be proportional to $(\Delta x)^2$. Fig. 16 shows the velocity at several points on the vertical centerline as a function of the grid size; the error is proportional to $(\Delta x)^2$ as expected. Results obtained by Ghia et al. [16] are also shown. We estimated the exact velocity by Richardson extrapolating the value to $\Delta x = 0$; the values obtained are a little smaller in absolute value than Ghia et al.'s values. The difference is largest near the driven lid, but is less than 1.5%. The reason for these difference is unknown.

Next we show results of the multigrid composite grid method for cavity flow at $Re = 1000$ on the 33×33 grid. The flow domain was decomposed into two subgrids of 25×33 grid nodes each, the ICMG method was used to solve the pressure equation with the convergence criterion mentioned above. In this case, the two grids coincide in the overlap region; the purpose of this case is merely to demonstrate the composite grid method and the method is essentially the one of Perng and Street [12]. Fig. 17(a) shows the velocity profile at steady state on the single grid, while Fig. 17(b) shows the velocity distribution found by the composite grid technique. In the single grid case, 2590 time steps were required to reach steady state with $\Delta t = 0.02$; 60 time steps require 1 minute of CPU time. For the composite grid case, 9 time steps require 1 minute of CPU time and 2562 iterations were needed to reach the steady state with $\Delta t = 0.02$. The difference is due to the composite grid having 1.5 times as many points as single grid and, in the single grid computations, only one V-cycle was allowed per one time step, while in the composite grid computations, we allowed five Schwarz iterations per time step. These two profiles agree and there is no discrepancy in the overlap region. Thus the accuracy of the scheme is confirmed.

7. APPLICATION TO COMPLEX FLOW FIELDS

Now we show some results for geometrically complex flows. First, we solved a lid-driven two-box cavity problem. The physical domain used is the same as in the 2-D two-box test computations (see Fig. 8). The edge length and lid speed are unity. The upper lid is driven toward the right while lower lid is driven toward the left. The Reynolds number is 1000. Each square had 33×33 nodes. The time step Δt was 0.02, making the Courant number less than 0.64 everywhere. In the computation of the pressure, the ICMG method with $\omega = 1.1$ was used and five Schwarz iterations with 1 V-cycle were allowed at each time step. In 1 minute, 5.6 time steps are taken on the CYDRA-5.

Fig. 18(a) shows the velocity profile at $t = 6$; there are two symmetric vortices. At $t = 9$, the two vortices have grown bigger and the centers closer, as shown in Fig. 18(b). Eventually the two vortices merge into one large vortex. The velocity profile at $t = 12$, shown in Fig. 18(c), contains one deformed large vortex. This vortex causes two large recirculating flow regions in the corners.

Fig. 18(d) shows the steady state velocity profile and Fig. 18(e) shows the corresponding vorticity distribution. Fig. 18(f) shows the pressure distribution; the pressure contours on the two grids differ slightly. The pressure is obtained by solving Eq (15); the surface integral of the RHS of Eq (15) on the overlap domain must be the same on the two subgrids, i.e.,

$$\int_{S_{\Omega_I}} \nabla\tilde{U}\, dS = \int_{S_{\Omega_{II}}} \nabla\tilde{U}\, dS \tag{38}$$

Since the scheme has second order spatial accuracy, Eq (38) must be obeyed to at least second order and the interpolations on the interior boundaries must have accuracy higher than second order. In the present work, we used the Coons patch method with second order peripheral functions and this is the probable cause of the small discrepancies in the pressure in the overlap region. The discrepancies are of the order of the accuracy of the scheme and are therefore not important.

Next we show results for an annular-box combined cavity flow. Fig. 19 shows the flow domain and grid. The Reynolds number is 1000 based on the length of square edge. The outer lid of the annular section was driven counterclockwise and the left edge of the square section moved vertically upward. Figs. 20(a) and (b) show the velocity and vorticity distributions at $t = 49$. Two large stable vortices occupy the flow domain. In the overlap domain, the velocities on the two grids agree very well. This shows that the usefulness of the composite grid strategy for geometrically complicated flows.

8. CONCLUSIONS

To simulate unsteady flows in geometrically complex domains, we discretized the 2-D unsteady Navier-Stokes equations using a staggered grid in curvilinear coordinates; the accuracy is second order.

The effectiveness of the composite multigrid approach for geometrically complex flows was demonstrated. It may be used both for flows in which an accurate time-history is required or for the computation of steady state flows. We investigated the convergence of the ICMG and CCMG methods for a 1-D problem and showed that acceleration of the interior boundary values is very effective. The optimal value of the acceleration parameter depends on boundary conditions. In the 2-D problem, the ICMG method again has better convergence than the CCMG method; however, acceleration of the ICMG method is also effective.

Finally, we simulated two complex cavity flows at $Re = 1000$ to demonstrate the effectiveness of the method for solving the Navier-Stokes equations in geometrically complex domains.

ACKNOWLEDGMENTS

This work was carried out during the first author's stay at Stanford University under the support of the Science and Technology Agency of Japan. He would like to express his thank to STA for giving him the chance to visit Stanford. The authors received useful suggestions through many discussions with Dr. Robert van der Wijngaart, Dr. Chin-Yuan Perng, Mr. Ramana Venkata and Prof. J. Oliger.

REFERENCES

(1) A. Brandt, "Multigrid Techniques: 1984 Guide, With Application to Fluid Dynamics", von Karman Institute Lecture Series, Brussels, 1984

(2) H. A. Schwarz, "Über einige Abbildungsaufgaben", J. Reine und Angewandte Mathematik, Vol.70, pp.105-120, 1869

(3) I. D. Faux and M. J. Pratt, "Computational Geometry for Design and Manufacture", Ellis Horwood Publishers, Chichester, pp. 198-203, 1979

(4) C. R. Maliska and G. D. Raithby, "A Method for Computing Three Dimensional Flows Using Non-Orthogonal Boundary-Fitted Coordinates", Int. J. Num. Meth. Fluids, Vol. 4, pp. 519-537, 1984

(5) B. P. Leonard, "A Stable and Accurate Convective Modelling Procedure Based on Quadratic Upstream Interpolation", Comp. Meth. Appl. Mech. Engr. 19, pp. 59-98, 1979

(6) R. W. MacCormack, "The Effect of Viscosity in Hypervelocity Impact Cratering", AIAA Paper, 69-354, 1969

(7) R. F. van der Wijngaart, "Composite-Grid Techniques and Adaptive Mesh Refinement in Computational Fluid Dynamics", Ph.D thesis, Stanford University, Jan. 1990

(8) P. L. Lions, "On the Schwarz Alternating Method III: A Variant for Nonoverlapping Subdomains", Proc. 3rd Int. Symp. Domain Decomposition Methods for Partial Differential Equations, SIAM, pp. 202-213, 1990

(9) R. L. Meakin and R. L. Street, "Simulation of Enviromental Flow Problems in Geometrically Complex Domains. Part 2: A Domain-Spliiting

Method", Comp. Meth. Appl. Mech. Engr. 68, pp. 311-331, 1988

(10) W. D. Henshaw and G. Chesshire, "Multigrid on Composite Meshes", SIAM J. Sci. Stat. Comp., Vol.8, No.6, pp. 914-923, Nov. 1987

(11) C. Y. Perng, "Adaptive-Multigrid Computations for Incompressible Flows, Including Geometry, Temperature, and Salinity Effects", Ph.D thesis, Stanford University, May, 1990

(12) C. Y. Perng and R. L. Street, "A Domain Decomposition Technique for Solving Geometrically Complex Flow Problems" Forum on Numerical Simulation of Convection in Electronic Equipment Cooling, ASME Winter Annual Meeting, Heat Transfer Div. Vol. 121, pp. 61-68, December 1989

(13) W. P. Tang, "Schwarz Splitting and Template Operators" CLaSSiC Report 87-3, Dept. of Computer Science Stanford University, 1987

(14) J. H. Ferziger, "Estimation and Reduction of Numerical Error", Forum on Methods of Estimating Uncertainty in Fluid Flow Computations, ASME Winter Annual Meeting, San Francisco, Dec, 1989

(15) G. D. Smith, "Numerical Solution of Partial Differential Equations: Finite Difference Methods", Oxford, Clarendon Press, (Third ed.), pp. 272-273, 1985

(16) U. Ghia, K. N. Ghia, and C. T. Shin, "High-Re Solutions for Incompressible Flow Using the Navier-Stokes Equations and a Multigrid Method", J. Comp. Phys., 48, pp. 387-411, 1982

LIST OF FIGURES

Fig. 13(b) Converged solution to the 2-D Poisson equation using ICMG

Fig. 14 Stencils for velocity nodes at the boundary

Fig. 15 Stencils for Coons patch interpolation of pressure and velocity

Fig. 16 Accuracy of lid-driven cavity flow results

Fig. 17(a) Velocity distribution on the centerline of the cavity flow using the non-composite grid, Re=1000

Fig. 17(b) Velocity distribution on the centerline of the cavity flow using the composite grid, Re=1000

Fig. 18(a) Velocity distribution in the two-box cavity flow, Re=1000, t=6

Fig. 18(b) Velocity distribution in the two-box cavity flow, Re=1000, t=9

Fig. 18(c) Velocity distribution in the two-box cavity flow, Re=1000, t=12

Fig. 18(d) Velocity distribution in the two-box cavity flow, Re=1000, t=60

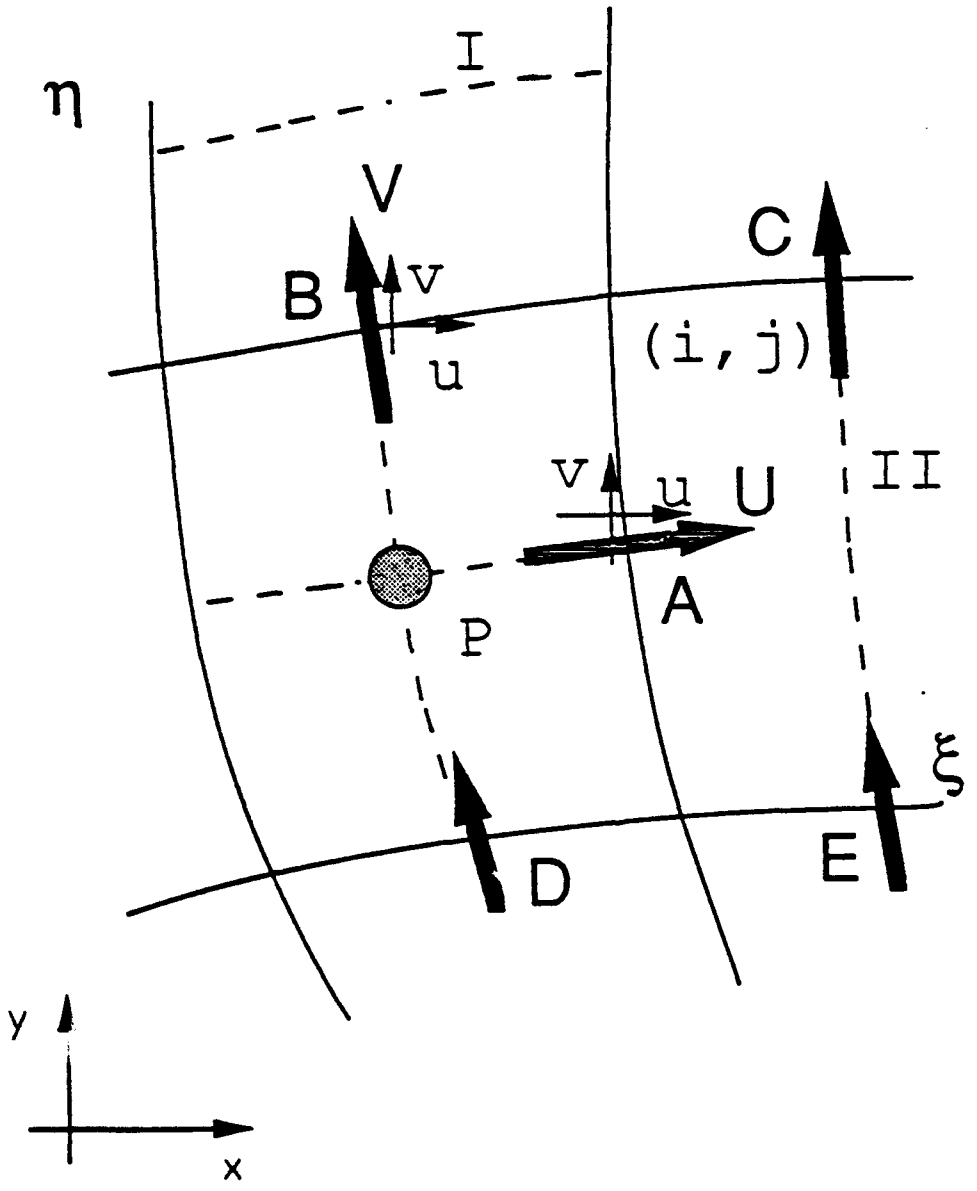
Fig. 18(e) Vorticity distribution in the two-box cavity flow, Re=1000, t=60, $\Delta\zeta = 1$, only $-20 \leq \zeta \leq 20$ is shown

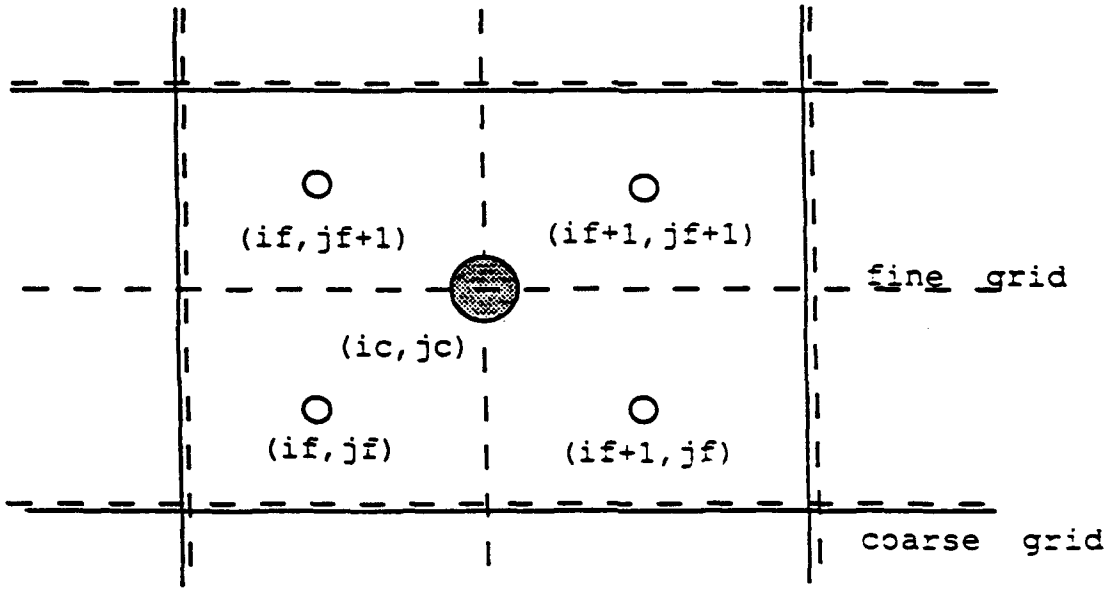
Fig. 18(f) Pressure contours in the two-box cavity flow, Re=1000, t=60, $\Delta p = 0.02$

Fig. 19 Flow domain and grid for annular-box cavity flow

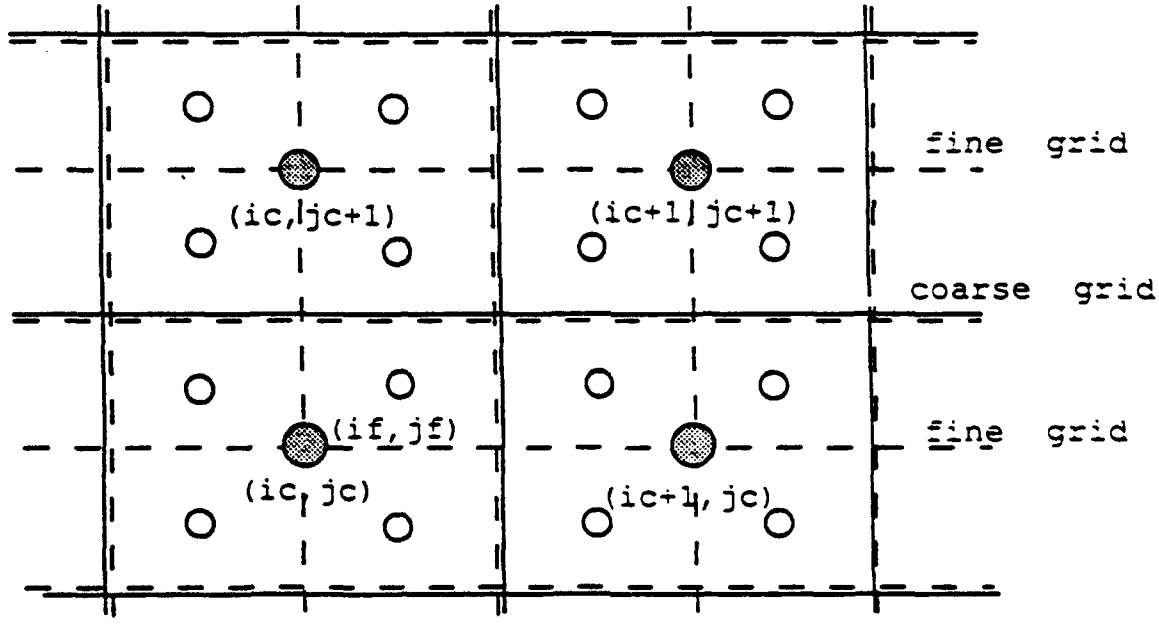
Fig. 20(a) Velocity distribution in annular-box cavity flow, Re=1000, t=49

Fig. 20(b) Vorticity distribution in annular-box cavity flow, Re=1000, t=49

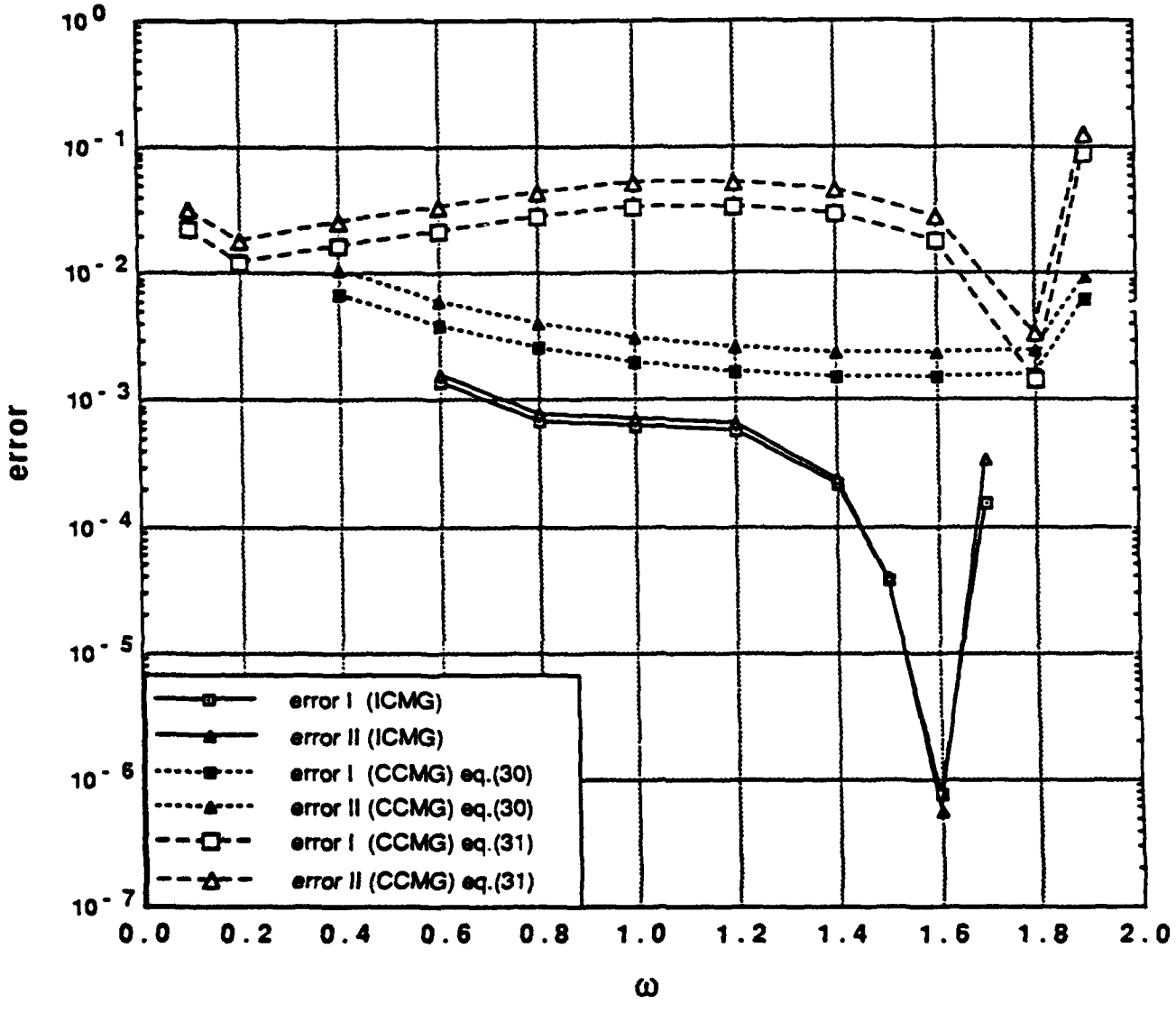

η
I
V
C
B
v
u
(i, j)
v
u
U
II
A
P
ξ
D
E
y
x

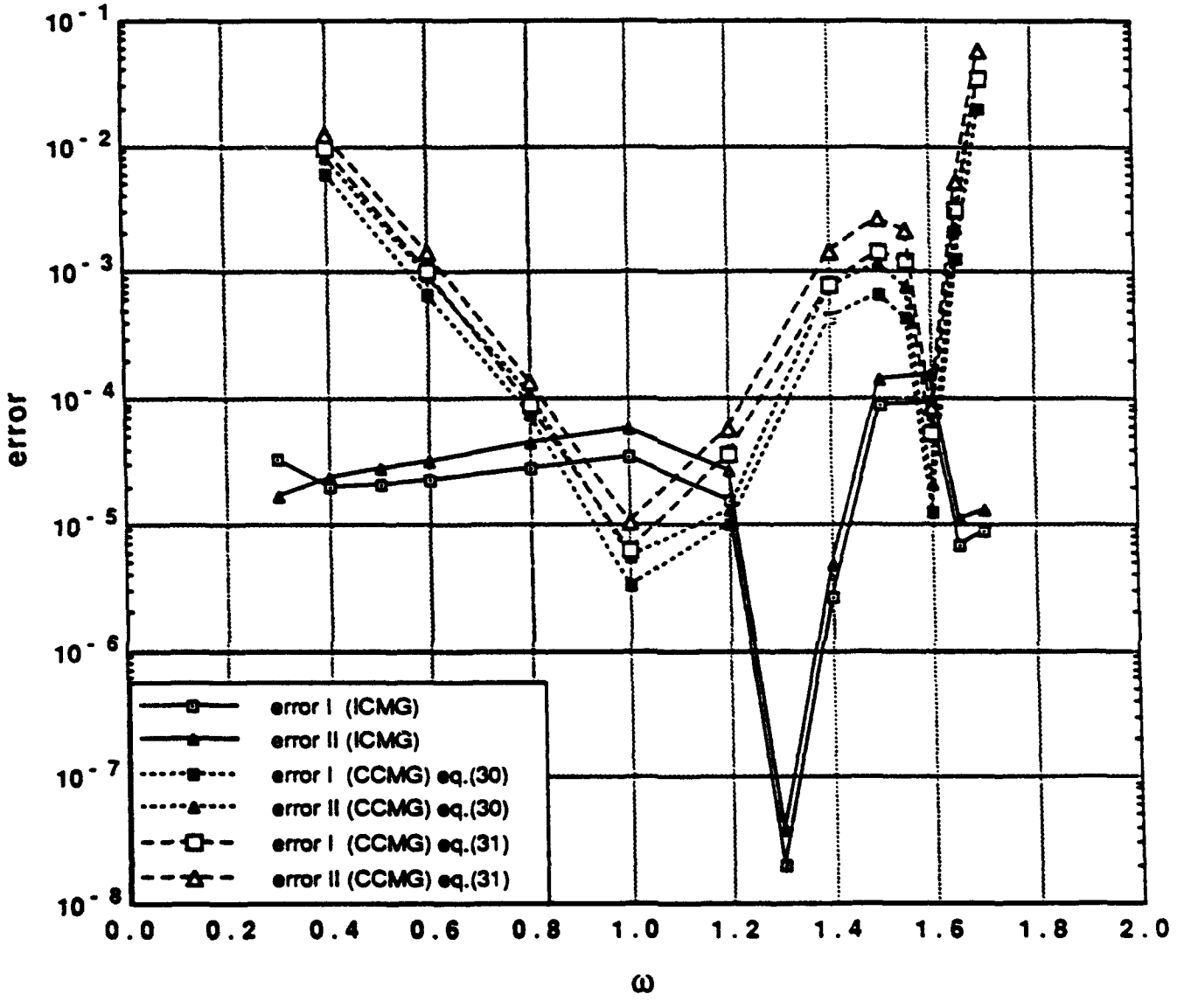


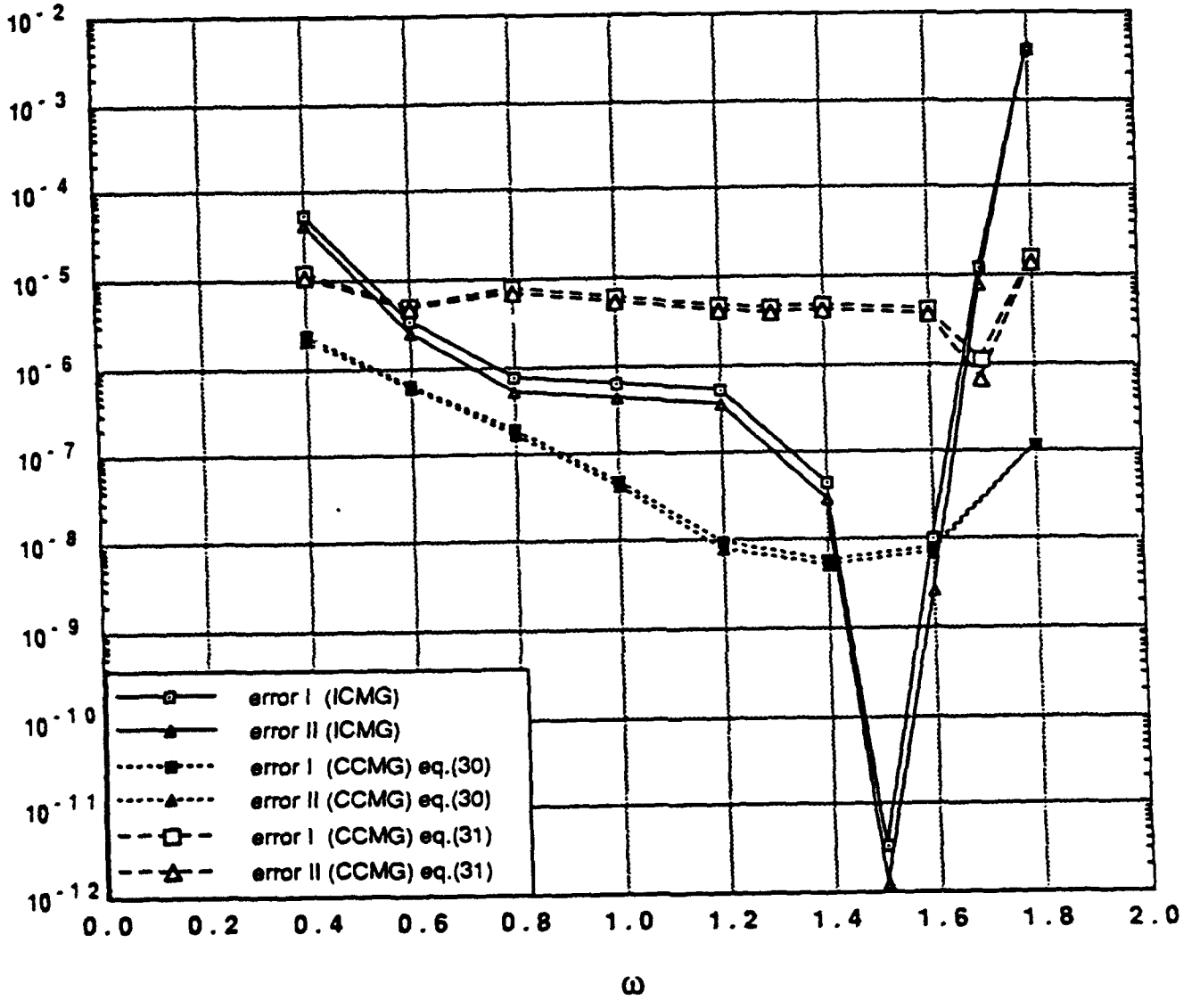
STENCIL FOR RESTRICTION

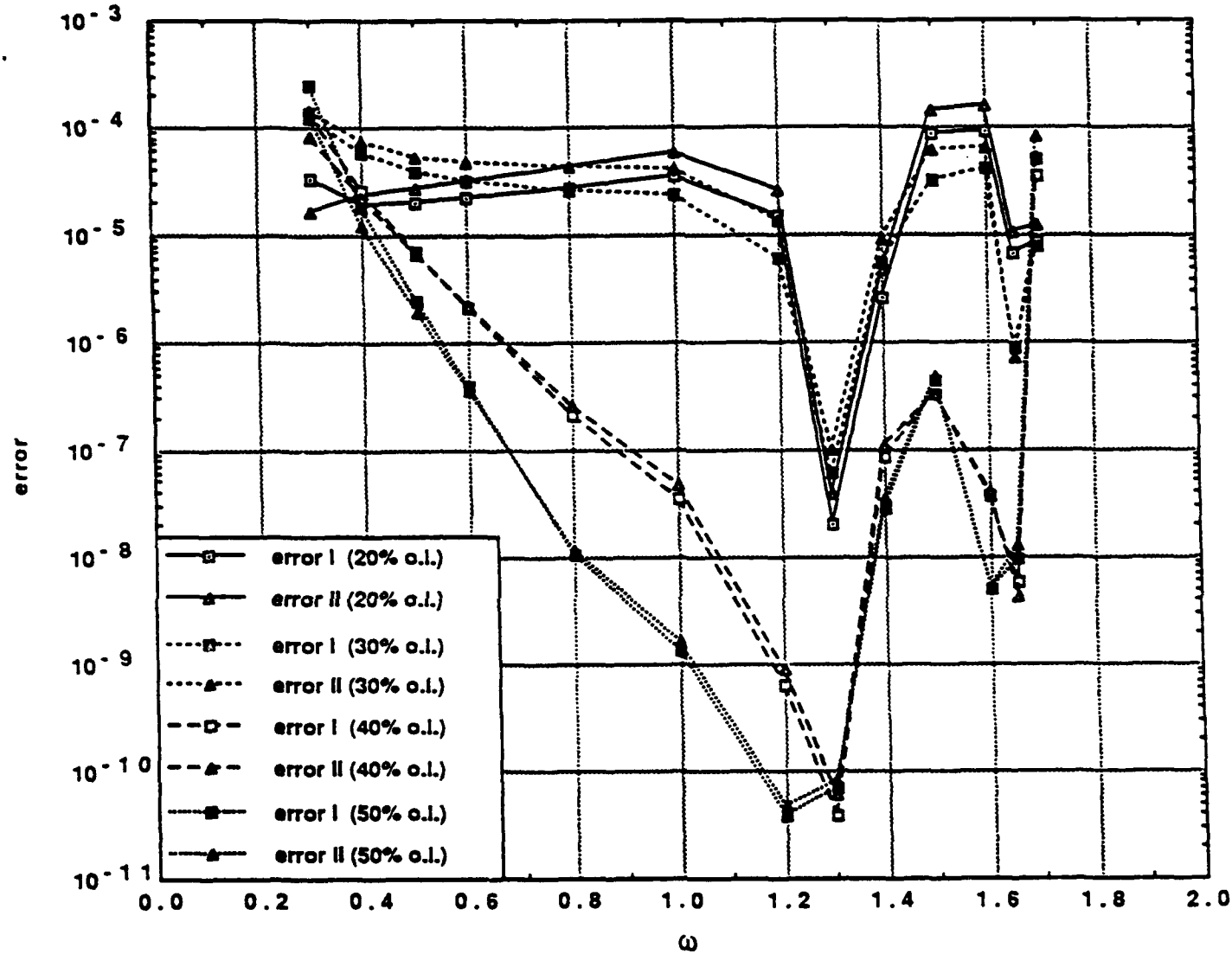


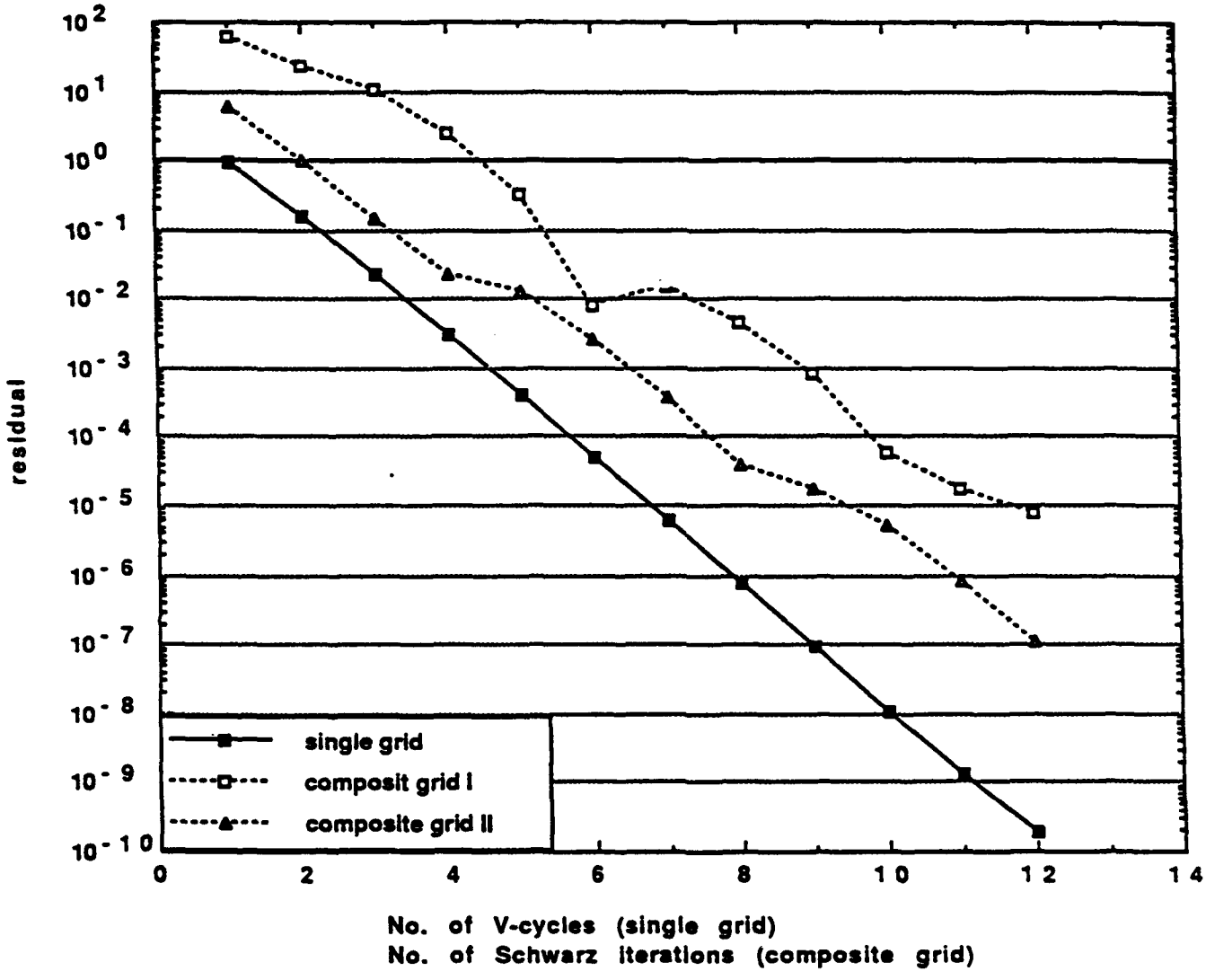
STENCIL FOR PROLONGATION

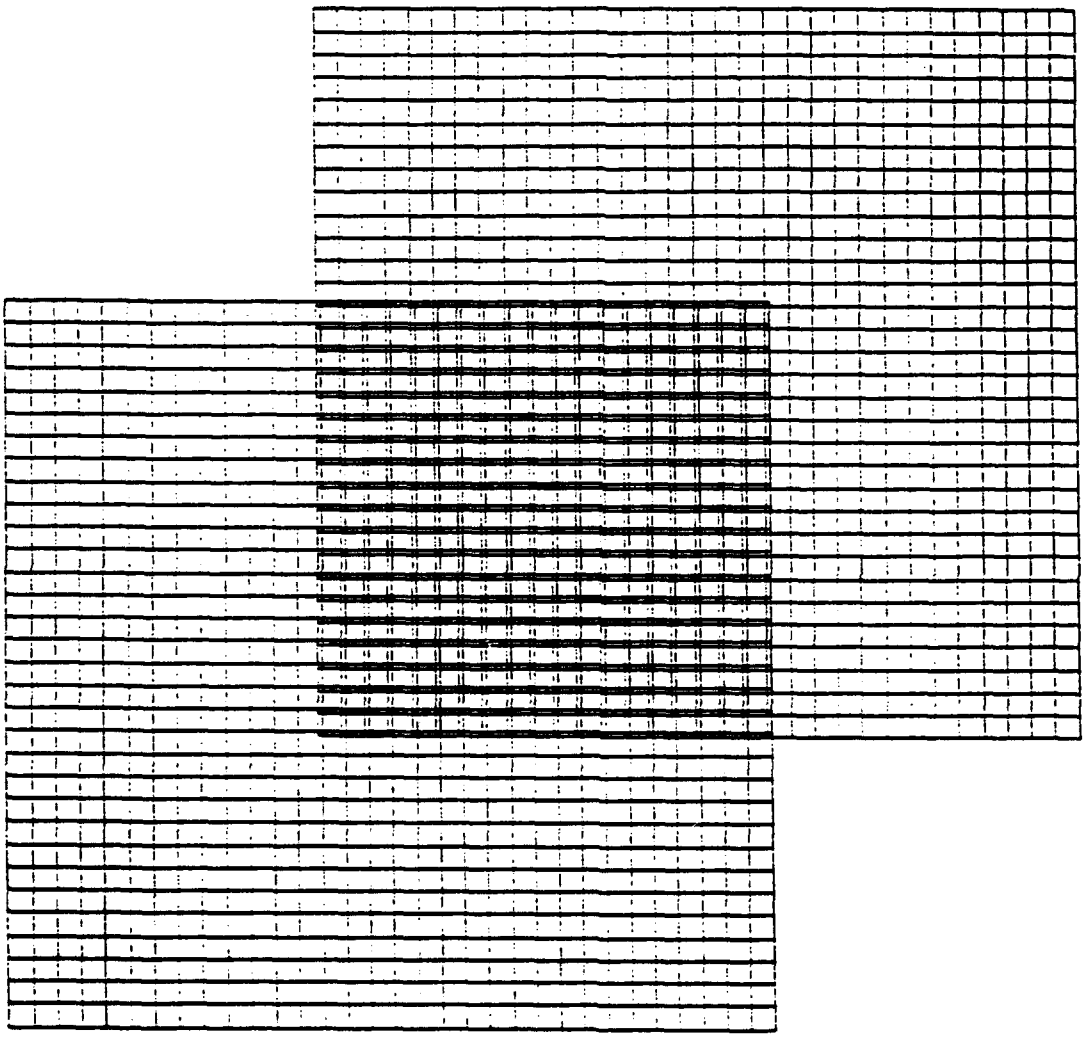

error
10^0
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1.8
2.0
ω
error I (ICMG)
error II (ICMG)
error I (CCMG) eq.(30)
error II (CCMG) eq.(30)
error I (CCMG) eq.(31)
error II (CCMG) eq.(31)


error
10 - 1
10 - 2
10 - 3
10 - 4
10 - 5
10 - 6
10 - 7
10 - 8
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1.8
2.0
ω
error I (ICMG)
error II (ICMG)
error I (CCMG) eq.(30)
error II (CCMG) eq.(30)
error I (CCMG) eq.(31)
error II (CCMG) eq.(31)


10 - 2
10 - 3
10 - 4
10 - 5
10 - 6
10 - 7
10 - 8
10 - 9
10 - 10
10 - 11
10 - 12
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1.8
2.0
ω
error I (ICMG)
error II (ICMG)
error I (CCMG) eq.(30)
error II (CCMG) eq.(30)
error I (CCMG) eq.(31)
error II (CCMG) eq.(31)


10 - 3
10 - 4
10 - 5
10 - 6
10 - 7
10 - 8
10 - 9
10 - 10
10 - 11
error
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1.8
2.0
ω
error I (20% o.l.)
error II (20% o.l.)
error I (30% o.l.)
error II (30% o.l.)
error I (40% o.l.)
error II (40% o.l.)
error I (50% o.l.)
error II (50% o.l.)


residual
10^{2}
10^{1}
10^{0}
10^{-1}
10^{-2}
10^{-3}
10^{-4}
10^{-5}
10^{-6}
10^{-7}
10^{-8}
10^{-9}
10^{-10}
0
2
4
6
8
10
12
14
single grid
composit grid I
composite grid II
No. of V-cycles (single grid)
No. of Schwarz iterations (composite grid)



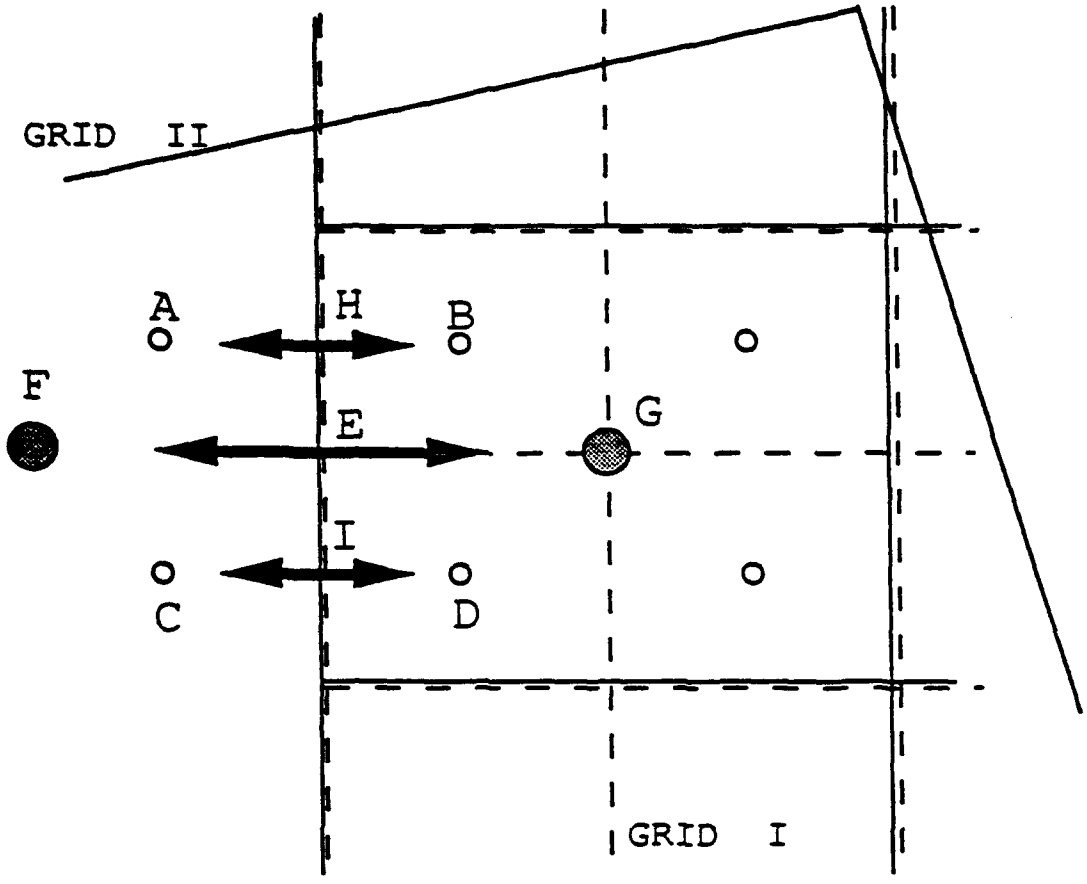
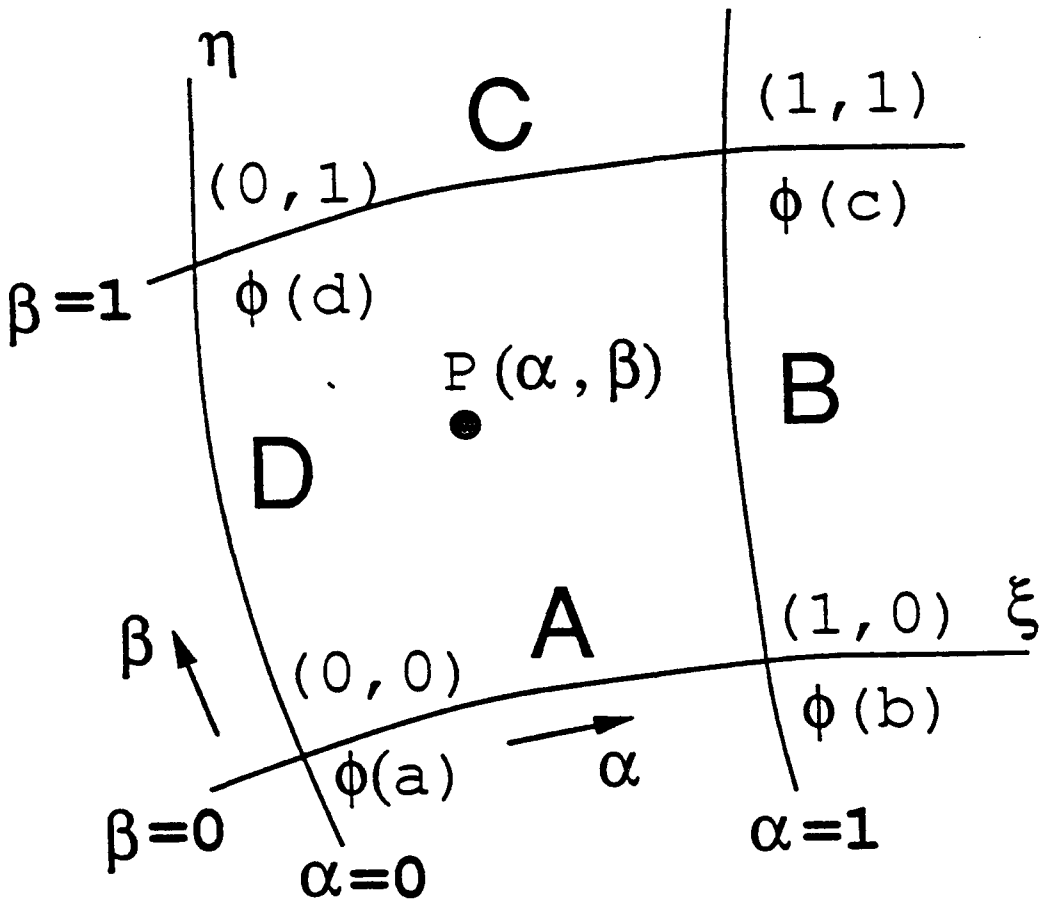
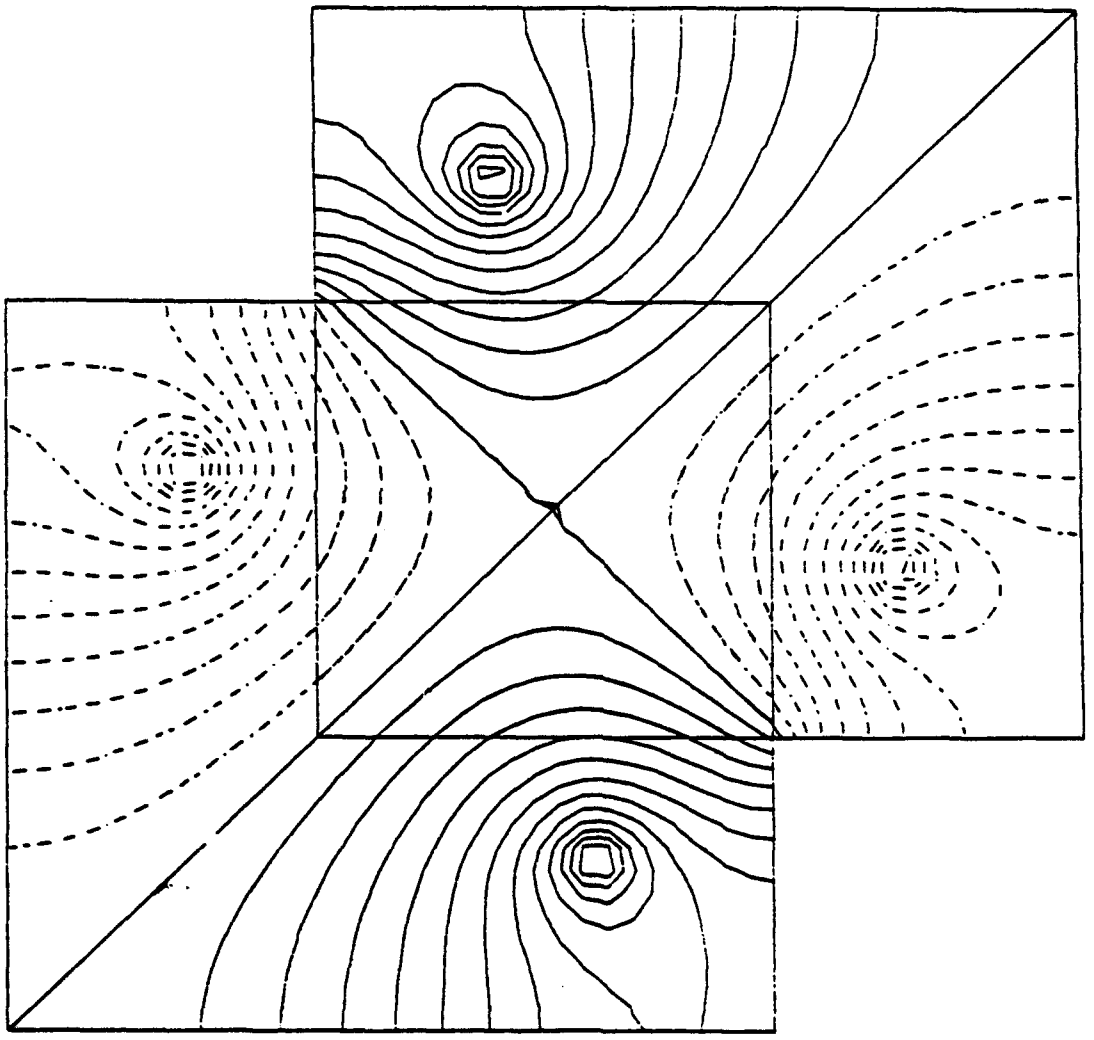
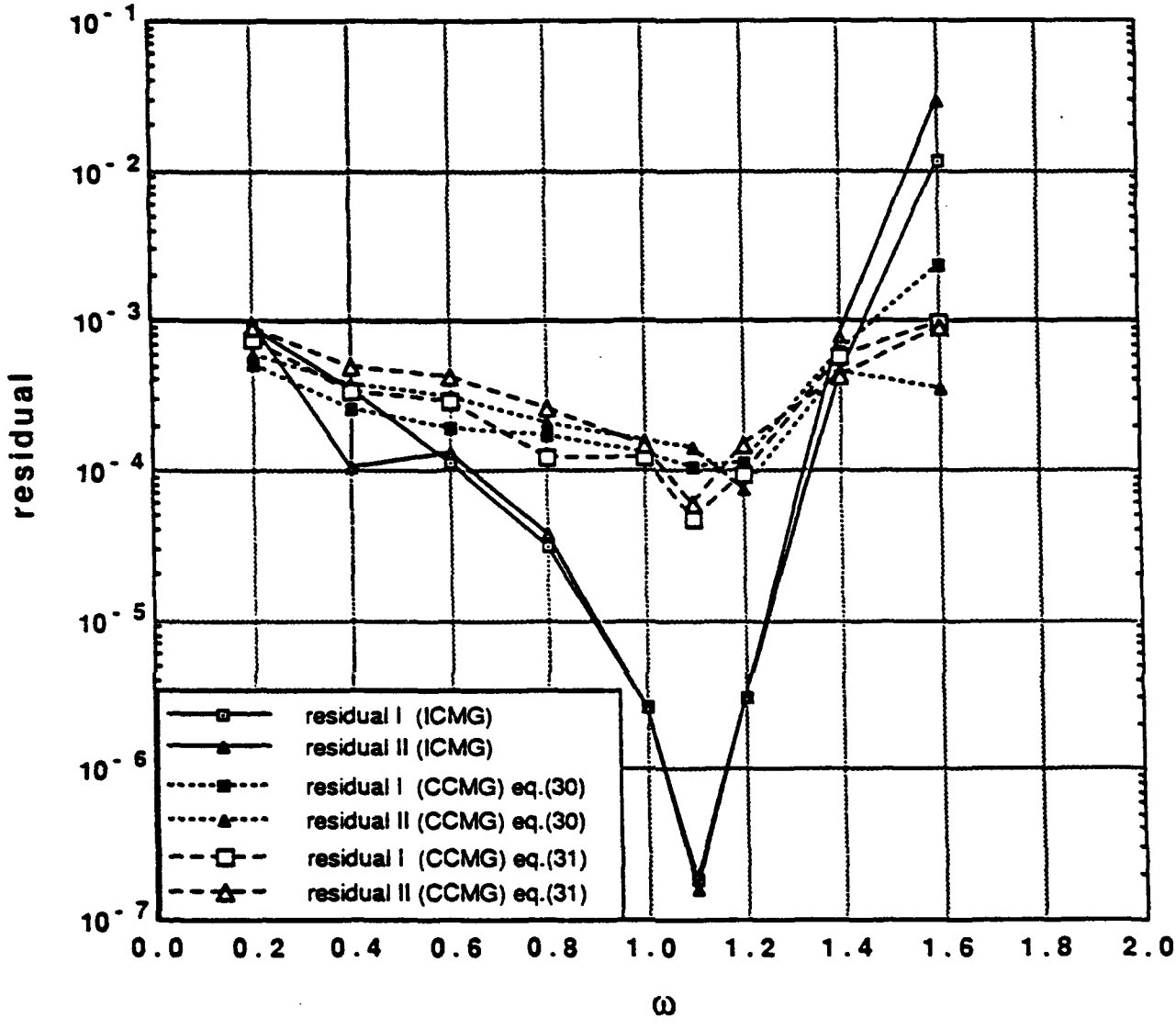
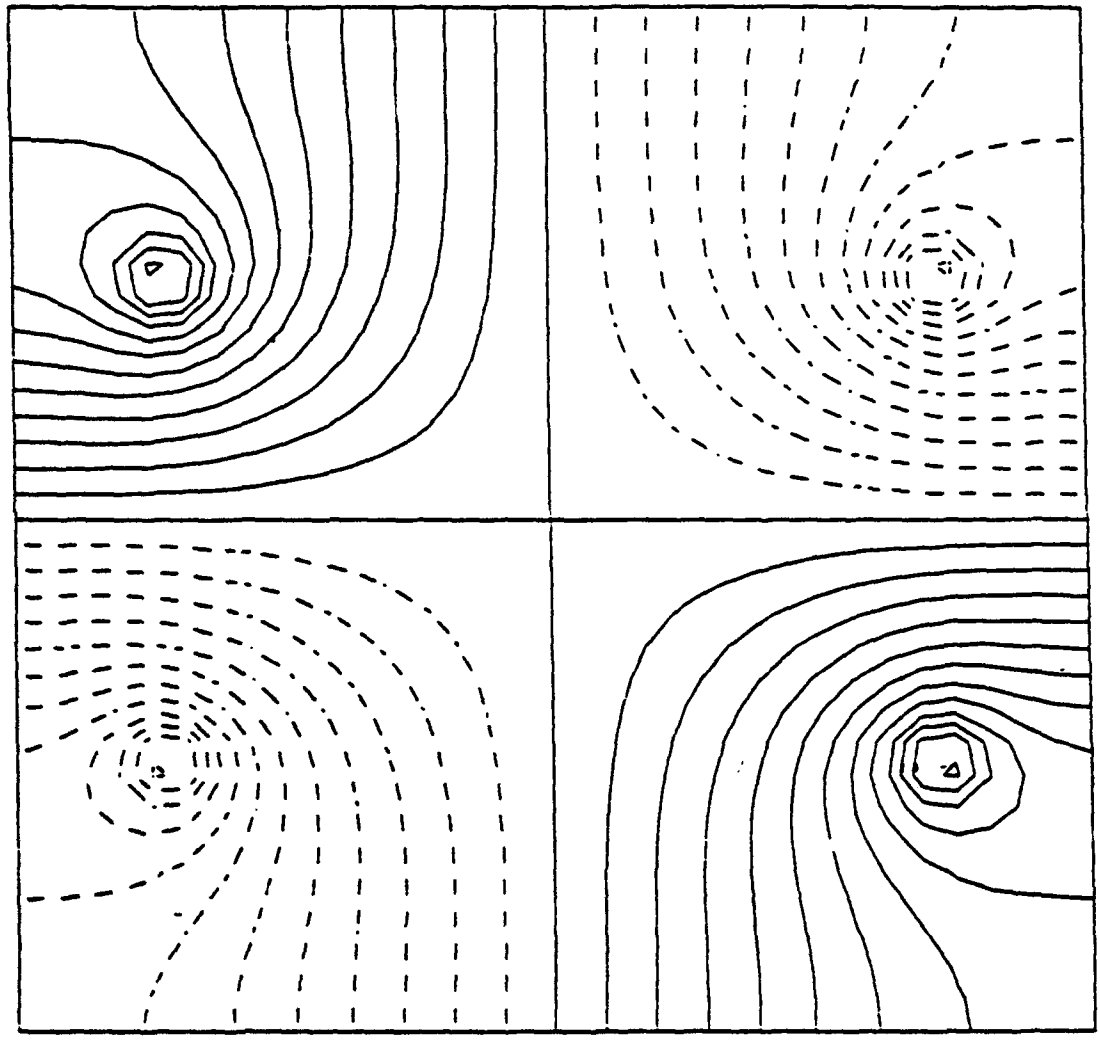


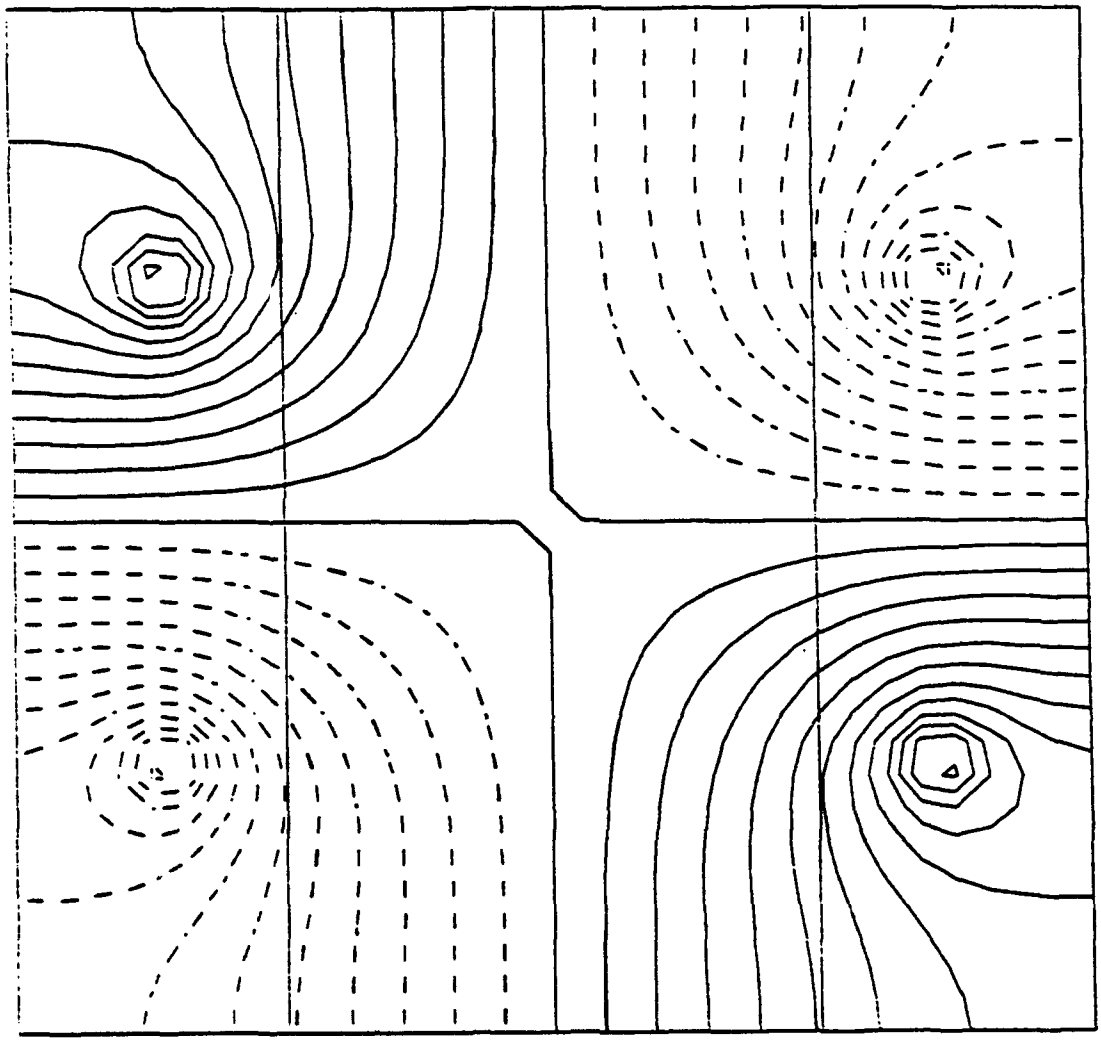
Fig. 9


η
C
(1,1)
(0,1)
φ(c)
β=1
φ(d)
P(α,β)
B
D
β
A
(1,0)
ξ
(0,0)
φ(b)
φ(a)
α
β=0
α=1
α=0

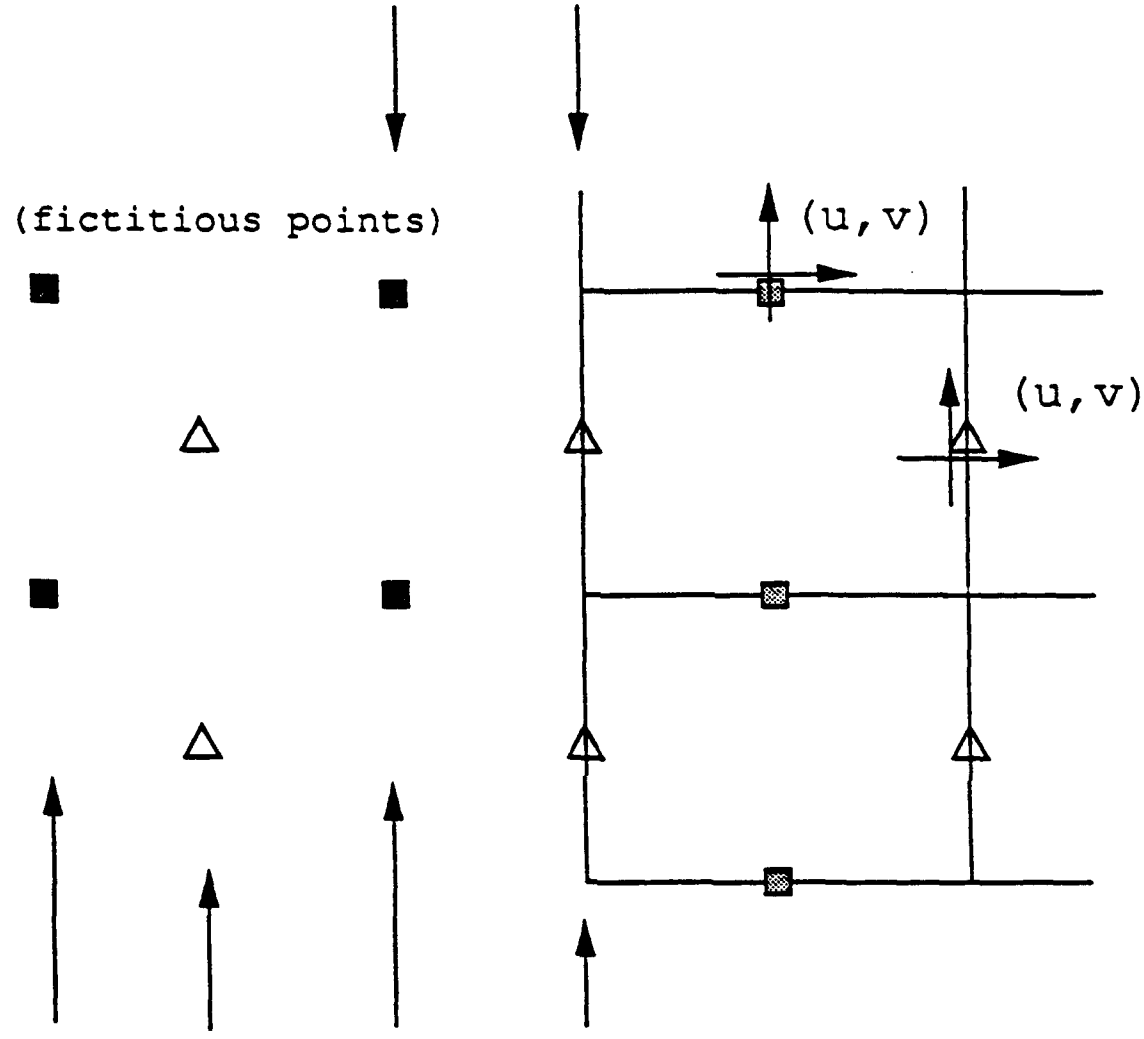



residual
10 - 1
10 - 2
10 - 3
10 - 4
10 - 5
10 - 6
10 - 7
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6
1.8
2.0
ω
residual I (ICMG)
residual II (ICMG)
residual I (CCMG) eq.(30)
residual II (CCMG) eq.(30)
residual I (CCMG) eq.(31)
residual II (CCMG) eq.(31)

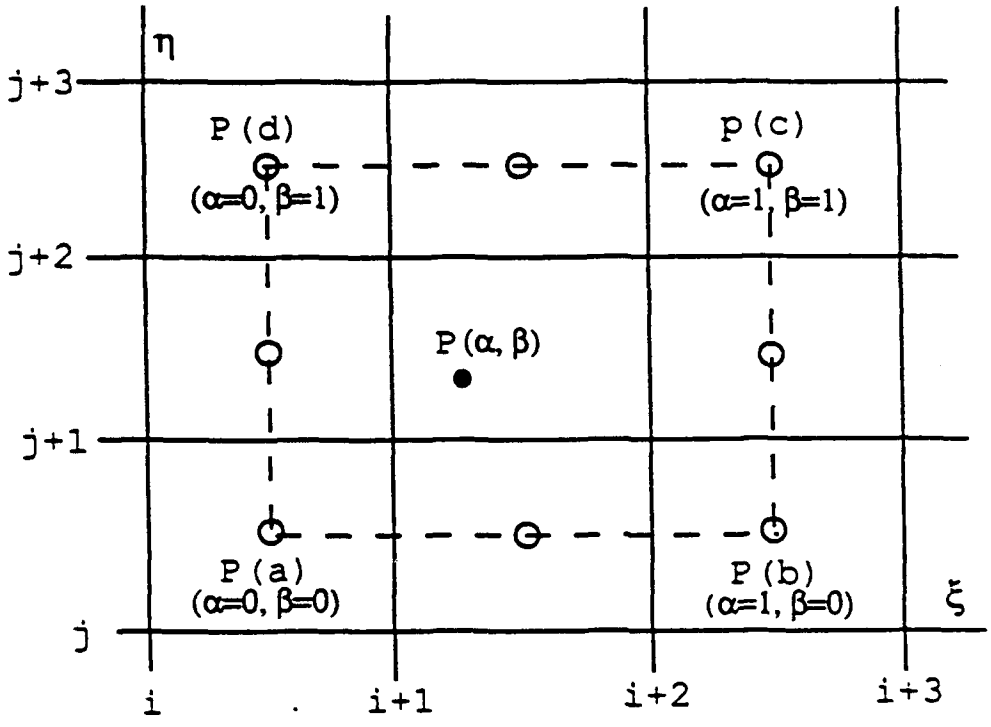




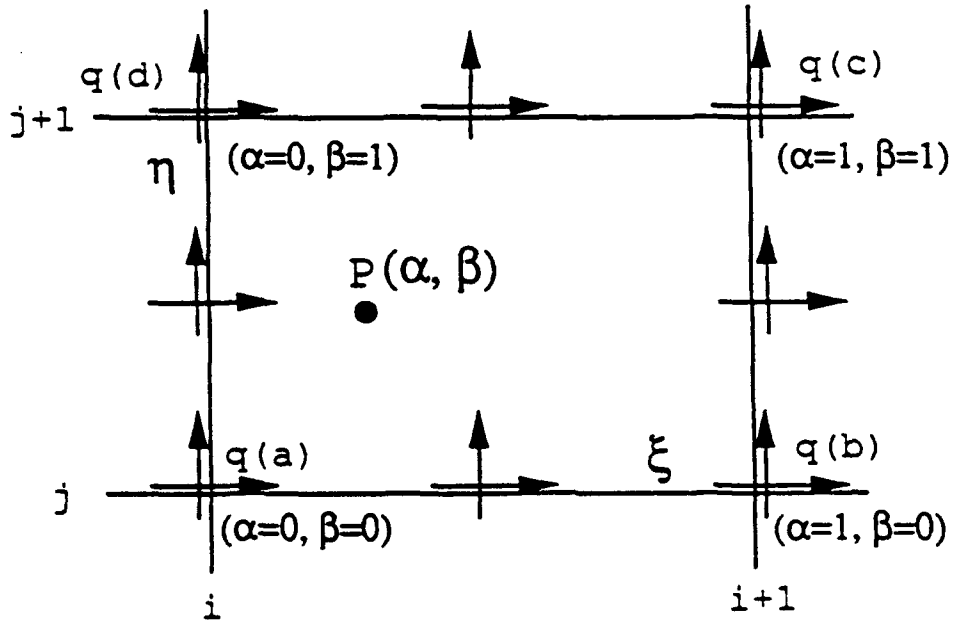
given by boundary conditions



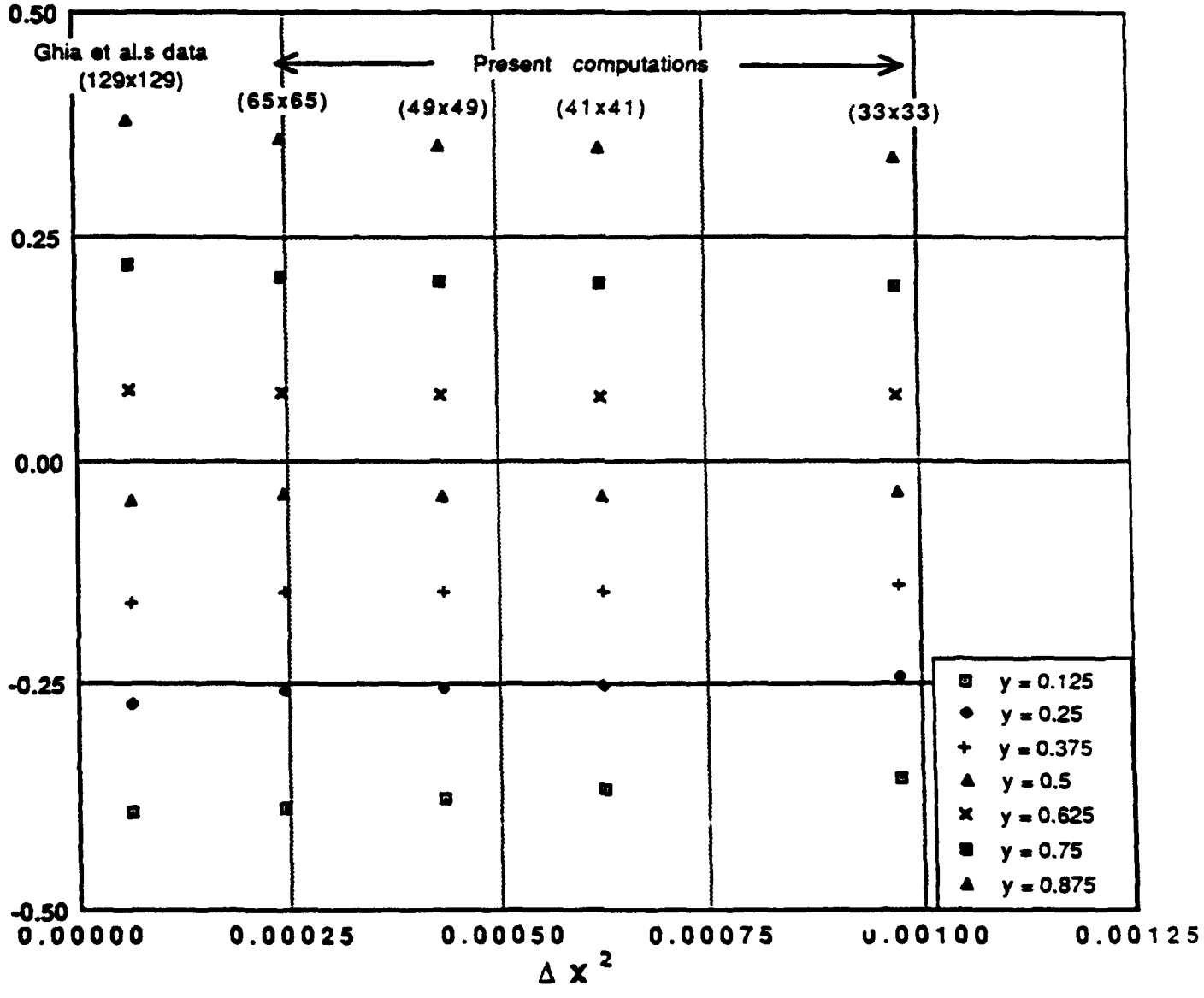
given by interpolation if included
in another subgrid



STENCIL FOR PRESSURE INTERPOLATION



STENCIL FOR VELOCITY INTERPOLATION


0.50
0.25
0.00
-0.25
-0.50
Ghia et al.s data
(129x129)
Present computations
(65x65)
(49x49)
(41x41)
(33x33)
y = 0.125
y = 0.25
y = 0.375
y = 0.5
y = 0.625
y = 0.75
y = 0.875
0.00000
0.00025
0.00050
0.00075
0.00100
0.00125
Δ X 2